

WEB SEARCH

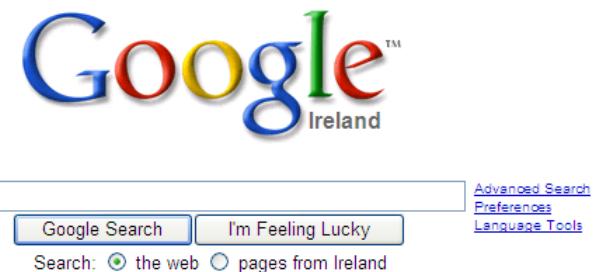
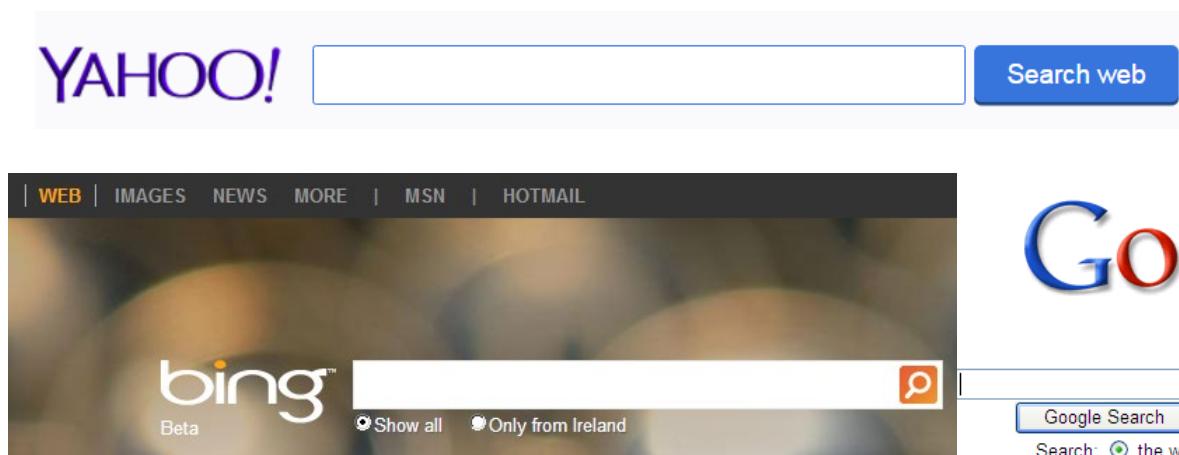
An Introduction

CT102
Information
Systems

WEB SEARCH ENGINES

Is Google your
usual web search
engine?

A web search engine is an online web information retrieval system that, given a query, returns a list of web pages that match a user's information need.



SEARCH ENGINE MARKET SHARE



GOOGLE'S DOMINANCE ...

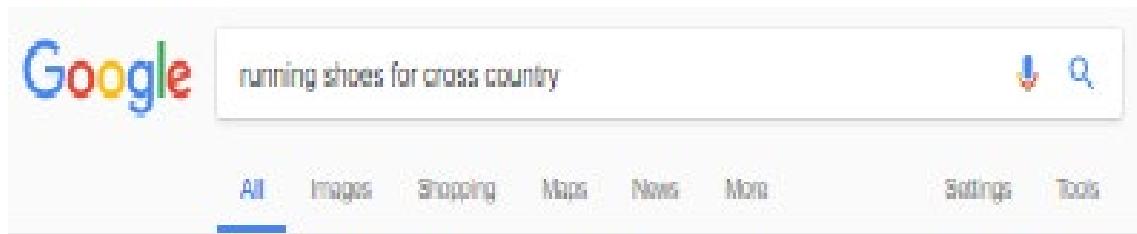
Worldwide, Google processes over 40,000 searches per second :

= approximately 3.5 billion searches per day

= approximately 1.2 trillion searches per year

CLASS QUESTION:

? How do search engines work ??

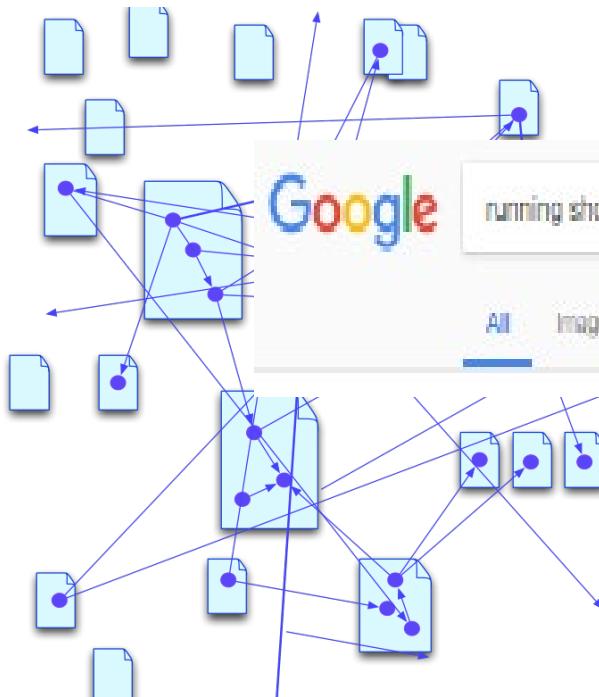


SEARCH ENGINE OVERVIEW:

Inputs, processing, outputs

Data → Information

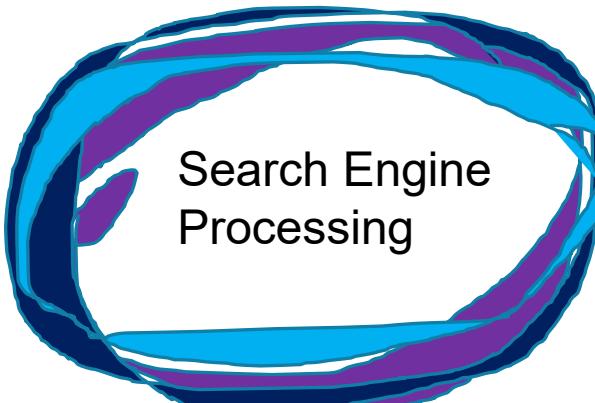
The data?



Input: The query

running shoes for cross country

All Images Shopping Maps News More



The output: The results

Google search results for "running shoes for cross country". The results include:

- ASICS™ IE: Running Shoes - ASICS™ Official Online Store
- www.sportsshoes.com ▾ Running Shoes Cross Country Spikes | SportsShoes.com
- View The Best Cross Country Shoes, Below.
- 1. Salomon Men's Speedcross 4 Trail. VIEW ON AMAZON ...
- 2. Saucony Men's Kilkenny XC5. VIEW ON AMAZON ...
- 3. New Balance Men's 900v4. VIEW ON AMAZON ...
- 4. New Balance Women's 700v5. ...
- 5. adidas Performance Women's XCS. ...
- 6. Saucony Women's Kilkenny XC5. ...
- 7. adidas Performance Men's XCS. ...
- 8. Women's Nike Zoom Rival XC.

More Items... • Aug 13, 2020

shoeadviser.com | athletic | best-cross-country-shoes
10 Best Cross Country Running Shoes - Shoe Adviser

About Featured Snippets Feedback

People also ask

What should I wear for cross country running? ▾

Can you use track shoes for cross country? ▾

DATA ON THE WEB

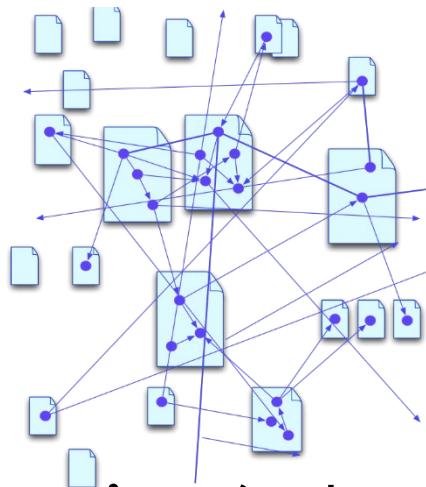
```
tic shows the global market share of leading internet search engines. In July 2
onth, Chinese search engine Baidu had a market share of 1.08 percent.

r>
earch in 1997, the worldwide market share of all search engines has been rather
re as of July 2018. The majority of Google revenues are generated through <a hr
as also expanded its services to mail, productivity tools, enterprise products,
googles-annual-global-revenue/">highest tech company revenues in 2017 with roug
  class="js-readingAid__gradient readingAid__gradient readingAid__gradient--invi
ute"
ReadSup" data-gtm="descriptionReadMore"
de"><span>Show more</span><i class="fa fa-caret-down margin-left-5"></i></span>
t--link link hideMobile"
tatistic216573" , "name": "Popup: Premium Account", "creative": "Global&#x20;ma
="paywall_c2a--sources--1"
          data-modal="#popupOverlay"
  data-file="sources" data-gtm="paywall_c2a--sources">
    </dd><dd href="/accounts/" class="text--link link"
tatistic216573" , "name": "Popup: Premium Account", "creative": "Global&#x20;ma
="paywall_c2a--publish"
          data-modal="#popupOverlay"
  data-file="publisher" data-gtm="showPublishLink">
    </dd><dt>Release date</dt><dd>

v id="info" class="tabContent tabContent--noPadding js_hidden"><dl><dt>Region</
e period</dt><dd>
0 to July 2018
roperties</dt><dd>

iv><div class=" actionBar float-right margin-top-10"><button id="statisticBrowse
 here to be redirected to see all your recently viewed statistics." data-toolti
="fa fa-history" aria-hidden="true"></i></button><button class="button button--
tatistic216573" , "name": "Popup: Premium Account", "creative": "Global&#x20;ma
```

DATA ON THE WEB



Typically **general-purpose web search engines** (such as Google) deal with data that:

- Has large portions of **unstructured** data.
- Has (potentially) some structure given by **html tags** indicating titles, sections, etc.
- Is often **natural language** data (e.g., English)
- Has (hyper)**links** to other web pages.
- Many search engines also use **semantically-tagged** data (e.g., dbpedia or similar).

STRUCTURED, SEMI-STRUCTURED AND UNSTRUCTURED DATA

Structured data: data that resides in a fixed field within a record or file, e.g., often relational (or other) database approach.

Semi-structured data: does not have a formal structure but does have tags or other information that convey meaning of data, e.g., XML or RDF documents with headings/sections, email, etc.

Unstructured data: data is not organised in any obviously meaningful way.

WHICH IS WHICH?

```
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
```

Fair Daffodils, we weep to see
You haste away so soon;
As yet the early-rising sun
Has not attain'd his noon.
Stay, stay, Until the hasting day
Has run But to the even-song;
And, having pray'd together, we
Will go with you along.
We have short time to stay, as you,
We have as short a spring;
As quick a growth to meet decay,
As you, or anything.
We die As your hours do, and dry Away,
Like to the summer's rain;

	employee										
	FName	MII	LName	SSN	BDATE	Address	SI	SALARY	SUPERSSI	DNO	
1	John	B	Smith	123456789	09/01/1965	731 Fondren, Houston, TX	M	€30,000.00	333445555	5	
2	Franklin	T	Wong	333445555	08/12/1955	638 Voss, Houston, TX	M	€40,000.00	888665555	5	
3	Joyce	A	English	453453453	31/07/1972	5631 Rice, Houston, TX	F	€25,000.00	333445555	5	
4	Ramesh	K	Narayan	666884444	15/09/1988	1000 Corporate Center Dr., Suite 550, Houston, TX	M	€35,000.00	333445555	5	
5	James	E	Borg	888665555	10/11/1989	1000 Corporate Center Dr., Suite 550, Houston, TX	M	€30,000.00	333445555	5	
6	Jennifer	S	Wallace	987654321	20/06/1990	1000 Corporate Center Dr., Suite 550, Houston, TX	M	€30,000.00	333445555	5	
7	Ahmad	V	Jabbar	987987987	29/03/1991	1000 Corporate Center Dr., Suite 550, Houston, TX	M	€30,000.00	333445555	5	
8	Alicia	J	Zelaya	999887777	19/07/1992	1000 Corporate Center Dr., Suite 550, Houston, TX	M	€30,000.00	333445555	5	

Structured, Semi-structured or Unstructured Data?

```
000000000000043300000300040000000000  
0043503400000000000000000000000000000000  
000000000030000000000000000000000000000000  
050000000000000000000000000000000000000000  
0040500000000200000000000000000000000000
```

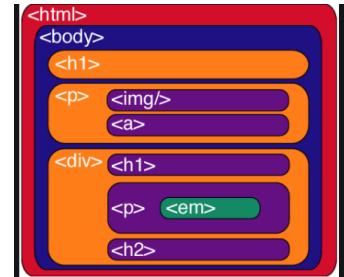
Structured, Semi-structured or Unstructured Data?

	FName	MII	LName	SSN	BDATE	Address	SI	SALARY	SUPERSSI	DNO
1	John	B	Smith	123456789	09/01/1965	731 Fondren, Houston, TX	M	€30,000.00	333445555	5
2	Franklin	T	Wong	333445555	08/12/1955	638 Voss, Houston, TX	M	€40,000.00	888665555	5
3	Joyce	A	English	453453453	31/07/1972	5631 Rice, Houston, TX	F	€25,000.00	333445555	5
4	Ramesh	K	Narayan	666884444	15/09/1962	975 Fire Oak, Humble, TX	M	€38,000.00	333445555	5
5	James	E	Borg	888665555	10/11/1937	450 Stone, Houston, TX	M	€55,000.00		1
6	Jennifer	S	Wallace	987654321	20/06/1941	291 Berry, Bellaire, TX	F	€43,000.00	888665555	4
7	Ahmad	V	Jabbar	987987987	29/03/1969	980 Dallas, Houston, TX	M	€25,000.00	987654321	4
8	Alicia	J	Zelaya	999887777	19/07/1966	3321 Castle, Spring, TX	F	€25,000.00	987654321	4

HTML Documents: Structured, Semi-structured or Unstructured Data?

```
<a name="d.en.83884"></a>
<div class="column-content richTextBlock">
<h3>Welcome Message from the Head of Computer Science</h3>
<p class="person"> The School of Computer Science is the largest academic discipline and one of the largest in NUI Galway overall. In 2019, QS World University Rankings put Computer Science & Information Systems @NUIG in 51st place. There are 22 academic staff and 51 research awards from Science Foundation Ireland, Irish Research Council, Enterprise Ireland, Health Research Board and other bodies. We have close to 700 students on our comprehensive suite of taught and research programmes at undergraduate and postgraduate level. We are actively engaged in a wide range of research topics in areas such as Artificial Intelligence; Machine Learning; Internet of Things; Image Processing; Simulation; Evolutionary Computation; and Informatics; Health Informatics, Energy Informatics, Enterprise Systems, Cyber-Security, Social Network Analysis, Digital Media and more.
<p>We have close to 700 students on our comprehensive suite of taught and research programmes at undergraduate and postgraduate level. We are actively engaged in a wide range of research topics in areas such as Artificial Intelligence; Machine Learning; Internet of Things; Image Processing; Simulation; Evolutionary Computation; and Informatics; Health Informatics, Energy Informatics, Enterprise Systems, Cyber-Security, Social Network Analysis, Digital Media and more.
<a href="/courses/undergraduate-courses/computer-science-and-information-technology.html">BSc in Computer Science</a>, <a href="/courses/postgraduate-courses/computer-science-and-information-technology.html">MSc in Computer Science</a>, <a href="/courses/undergraduate-courses/information-technology.html">BSc in Information Technology</a>, <a href="/courses/postgraduate-courses/information-technology.html">MSc in Information Technology</a>, <a href="/courses/undergraduate-courses/software-engineering.html">BSc in Software Engineering</a>, <a href="/courses/postgraduate-courses/software-engineering.html">MSc in Software Engineering</a>, <a href="/courses/undergraduate-courses/digital-media.html">BSc in Digital Media</a>, <a href="/courses/postgraduate-courses/digital-media.html">MSc in Digital Media</a>, <a href="/courses/undergraduate-courses/electronic-and-computer-engineering.html">BE in Electronic and Computer Engineering</a>, <a href="/courses/postgraduate-courses/electronic-and-computer-engineering.html">MEng in Electronic and Computer Engineering</a>, <a href="/courses/undergraduate-courses/science-undenominated.html">Bachelor of Science</a>, and the MA in Digital Media (with the <a href="http://www.filmschool.ie/filmschool/">Hustle and Hustle</a>). In addition to our research and teaching, our staff and students are heavily engaged with the community (free introductory computer classes for the digitally excluded), CoderDojo (an international movement of hackerspaces) and Galway Games Group. We are located in the Information Technology Building, a dedicated 4100 square metre building in midtown Galway, with state-of-the-art facilities for our students on taught programmes, dedicated research office space for our researchers, and a makerspace.
```

Characteristics of HTML files

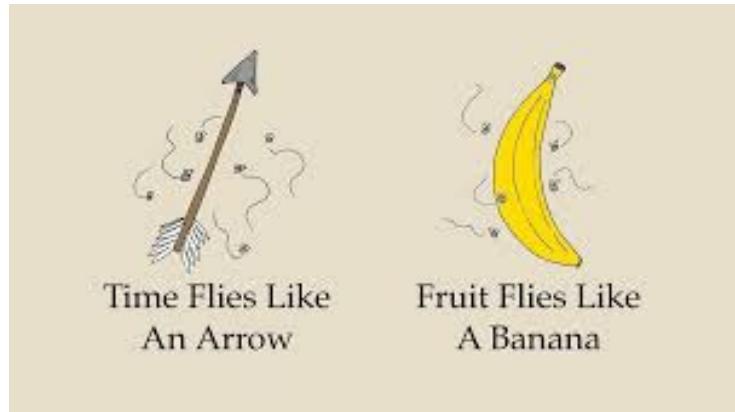


- The files can contain natural language text, audio, images, video, etc.
- HTML tags define the format of the content (headings, bullet points, etc.) From these tags we can sometimes infer importance of certain text, e.g. `<title>` indicates the title and if correct will give the words that are probably most important in the page but they do not give us *meaning*.
- HTML files contain A LOT of formatting tags.
- One important HTML is the `href` tag
- HTML files are displayed/rendered by a browser but this is not the view a program (spider or scraper) sees – it sees the raw HTML file.

Go to your favourite web page, right click on page and choose ‘View page source’

Natural language is generally *unstructured* and meaning is not easy to determine

- Writing programs to “read”/process, decipher, “understand” and make sense of (analyse) human languages is a difficult task.
- “Language is compositional”, i.e., letters form words, words form phrases and sentences.
- The meaning of a phrase can be “larger” than the individual words that comprise it.



Fair Daffodils, we weep to see
You haste away so soon;
As yet the early-rising sun
Has not attain'd his noon.
Stay, stay, Until the hasting day
Has run But to the even-song;
And, having pray'd together, we
Will go with you along.

Natural language is generally *unstructured* and meaning is not easy to determine

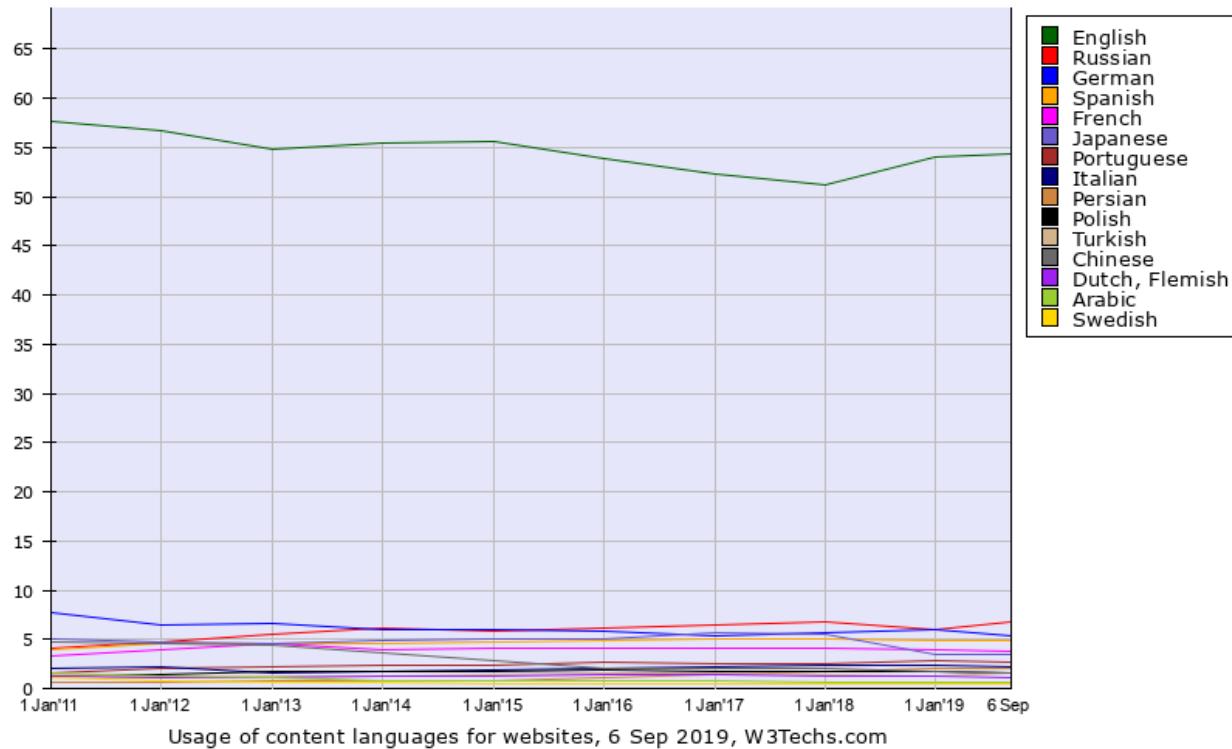
- Recent success has been due to “deep learning” (machine learning) techniques that learn from huge amounts of data.
- Many existing techniques use *statistical approaches* which infer meaning from frequencies of letters, symbols, words, etc.

? Which language is most predominantly used for HTML text content?

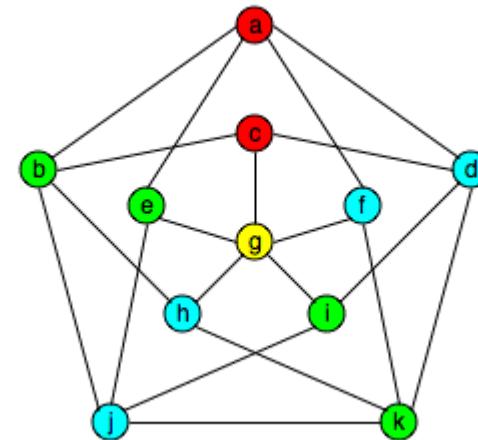
LANGUAGES USED FOR WEBSITES

(https://w3techs.com/technologies/history_overview/content_language/ms/y)

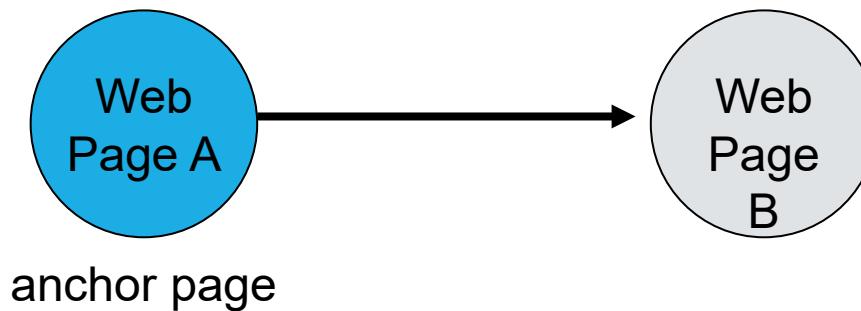
The diagram shows only content languages with more than 1% usage.



ANOTHER IMPORTANT ASPECT OF HTML DOCUMENTS: Linked Data



Can view the static Web as consisting of static HTML pages (containing text, images, video etc.) and **in addition** the hyperlinks between pages



e.g. page A has HTML:

<u href = <http://www.nuigalway.ie>> NUI, Galway
where B = <http://www.nuigalway.ie>

FROM OUR WEBSITE . . .

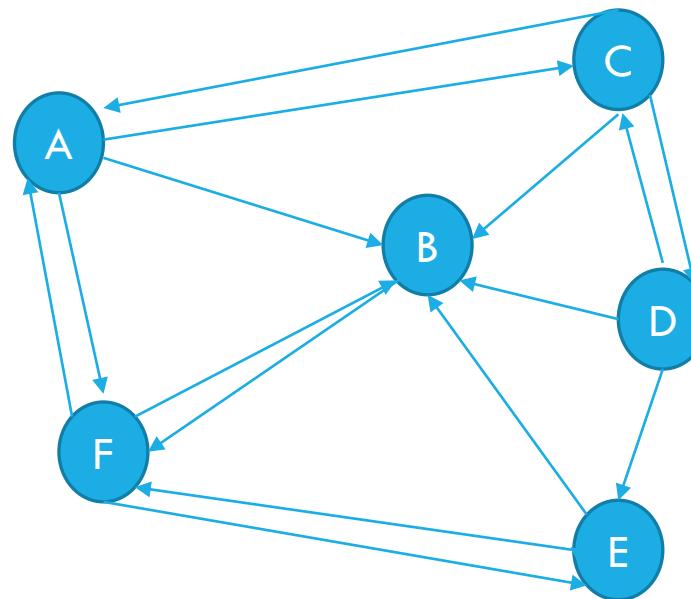
Our undergraduate taught programmes include the [BSc in Computer Science & Information Technology](/courses/undergraduate-courses/computer-science-and-information-technology.html) and the [Bachelor of Arts with Information Technology](/engineering-informatics/information-technology/programmes/undergraduateprogrammes/itasasubjectforartsstudents/). Our postgraduate conversion programmes include the

andnbsp; Science and Engineering and Informatics and one of the largest in NUI Galway overall. In 2017 the QS World University Rankings put Computer Science & Information Systems @NUIG internationally in the 201-250 bracket, and second in Ireland.</p><p>>>There are 22 academic staff and 5 technical & administrative staff in Information Technology. We also have over 80 full-time researchers at M.Sc., Ph.D. and postdoctoral level. We are actively engaged in a wide range of research topics in areas such as Artificial Intelligence; Machine Learning; Human-Computer Interaction; Medical Informatics; Networks & Communications; Internet of Things; Image Processing; Simulation; Evolutionary Computation; and Information Retrieval. We apply our research expertise in application areas such as Health Informatics, Energy Informatics, Enterprise Systems, Cyber-Security, Social Network Analysis, Digital Media, and Games. Our research programmes are funded by competitive research awards from Science Foundation Ireland, Irish Research Council, Enterprise Ireland, Health Research Board and the European Union’s Horizon 2020 Programme.</p><p>>>We have close to 700 students on our comprehensive suite of taught and research programmes at undergraduate and postgraduate level. Our undergraduate taught programmes include the BSc in Computer Science & Information Technology and the Bachelor of Arts with Information Technology. Our postgraduate conversion programmes include the MSc in Software Design and Development, the Higher Diploma in Software Design and Development, the Higher Diploma in Software Design and Development (Industry

THE WEB GRAPH

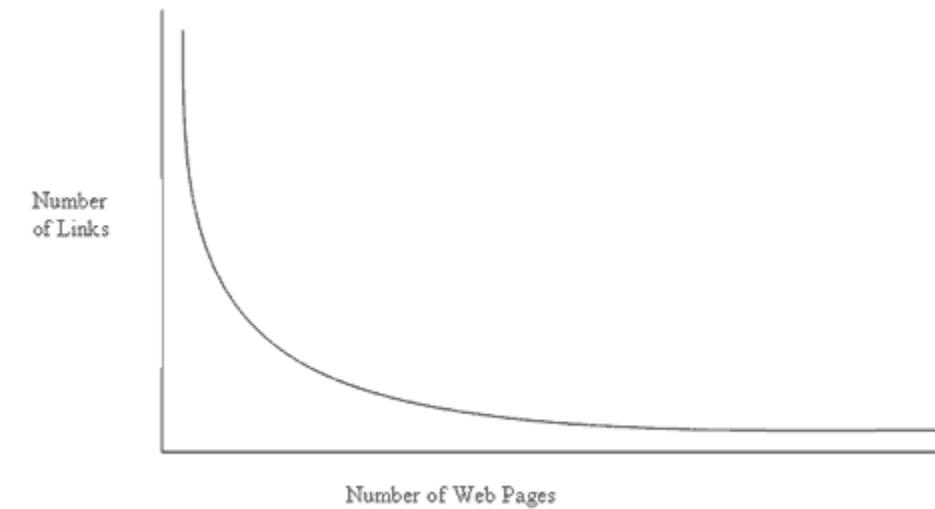
The hyperlink connections between pages can be viewed as a ([directed](#)) graph

An example of a web graph representation of 6 web pages



WEB LINK DISTRIBUTION

- Web page links are not randomly distributed.
- Distribution is widely reported to be a power law , in which the total number of web pages with in-degree i is proportional to $1/i^c$ (c a constant)
- i.e. only a small portion of web pages have a huge number of links



So we have now discussed “text data on the web” ... very important to understand what we are working with

Typically **general-purpose web search engines** (such as Google) deal with data that:

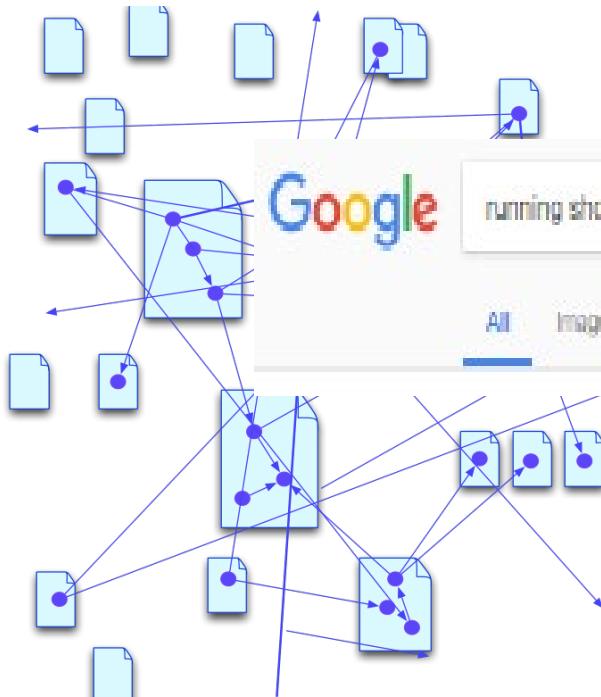
- Has large portions of **unstructured** data ✓
- Has (potentially) some structure given by **html tags** indicating titles, sections, etc. ✓
- Is often **natural language** data (e.g., English) ✓
- Has (hyper)**links** to other web pages ✓
- ~~Many search engines also use **semantically-tagged** data (e.g., dbpedia or similar)~~ ✗

SEARCH ENGINE OVERVIEW:

Inputs, processing, outputs

Data → Information

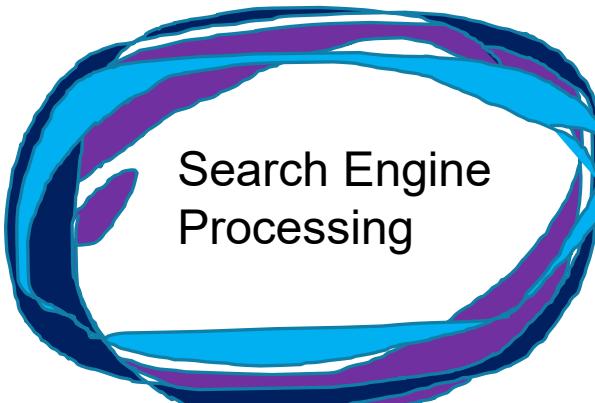
The data?



Input: The query

running shoes for cross country

All Images Shopping Maps News More



The output: The results

Google search results for "running shoes for cross country". The results include:

- ASICS™ IE: Running Shoes - ASICS™ Official Online Store
- www.sportsshoes.com ▾ Running Shoes Cross Country Spikes | SportsShoes.com
- View The Best Cross Country Shoes, Below.
- 1. Salomon Men's Speedcross 4 Trail. VIEW ON AMAZON ...
- 2. Saucony Men's Kilkenny XC5. VIEW ON AMAZON ...
- 3. New Balance Men's 900v4. VIEW ON AMAZON ...
- 4. New Balance Women's 700v5. ...
- 5. adidas Performance Women's XCS. ...
- 6. Saucony Women's Kilkenny XC5. ...
- 7. adidas Performance Men's XCS. ...
- 8. Women's Nike Zoom Rival XC.

More Items... • Aug 13, 2020

shoeadviser.com | athletic | best-cross-country-shoes
10 Best Cross Country Running Shoes - Shoe Adviser

About Featured Snippets Feedback

People also ask

What should I wear for cross country running?

Can you use track shoes for cross country?

Let's look at the inputs and outputs next....



running shoes for cross country

X |

All Images Shopping News Videos More

Settings Tools

About 94,100,000 results (0.53 seconds)

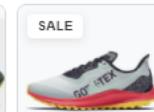
Ads · See running shoes for cross country



Asics Gel Fujitrabuco 7...
€89.59
SportsShoes.com
 (103)
By RedBrain



Brooks Mach 19 Running Shoe...
€80.00
Brooks Running
 (3)
By Google



Nike Air Zoom Pegasus 36 Tr...
€105.47 €150
Nike Official
Free delivery
By PriceSearcher



New Balance Mens MT590 ...
€41.95
MandMDirect.ie
By Shoptail



On Cloudventuro Waterproof...
€179.95
On
 (114)
By Google

www.asics.com/ ▾

ASICS™ IE: Running Shoes - ASICS™ Official Online Store

Free delivery and free returns on all orders. Official range available here. Shop genderless **running shoes** at ASICS™ online. Extended range. New collections. Online exclusives.

www.sportsshoes.com/ ▾

Running Shoes Cross Country Spikes | SportsShoes.com

Shop our range of **Cross Country Shoes** at SportsShoes.com. We're experts in what we do.

View The Best Cross Country Shoes, Below.

1. Salomon Men's Speedcross 4 Trail. [VIEW ON AMAZON](#). ...
2. Saucony Men's Kilkenny XC5. [VIEW ON AMAZON](#). ...
3. New Balance Men's 900v4. [VIEW ON AMAZON](#). ...
4. New Balance Women's 700v5. ...
5. **adidas Performance** Women's XCS. ...
6. Saucony Women's Kilkenny XC5. ...
7. **adidas Performance** Men's XCS. ...
8. Women's Nike Zoom Rival XC.

[More items...](#) • Aug 13, 2020

shoeadviser.com > athletic > best-cross-country-shoes

10 Best Cross Country Running Shoes - Shoe Adviser

Input: generally, web search begins with an “information need” ... what is this?

- Information needs are related to **problems**
- Part of “Information seeking behaviour” that a person will engage in given some problem
- The process of “asking” a question of an information system
- Often non-trivial to map or translate an information need in to a query

information need -> query

CLASS WORK ... IN GROUPS

- > Pick a query from the list
- > Use different search engines/devices
- > Look at results – compare across search engines/devices
- > Discuss what you think is happening

today's weather forecast

pizza delivery

5G threats

side-effects of covid vaccines

covid-19 in galway

emploi d'été en france

best route to Rosslare from Galway

PROPERTIES OF INFORMATION NEEDS

May be well-defined

May be vague

May contain no text (e.g. an image or tune)

A single correct answer may not exist

The answer may be surprising or not

The answer may be believable or not

Many solutions may match an information need, but a user's tastes and preferences may be the deciding factor in which solution the user deems relevant.

Now considering the outputs

- Lots of results!
- Different types of results
- Depending on search engine chosen could get many ads for some queries
- Lists of different websites and images/description

running shoes for cross country

About 94,100,000 results (0.53 seconds)

Ads See running shoes for cross country

Asics Gel Fujitrabuco 7 ...
€89.59
SportsShoes.com
★★★★★ (103)
By RedBrain

Brooks Mach 19 Running Shoe...
€80.00
Brooks Running
★★★★★ (3)
By Google

Nike Air Zoom Pegasus 36 Tr...
€105.47 €150
Nike Official
Free delivery
By Pricemart...

SALE New Balance Mens MT590 ...
€41.95
MandMDirect.ie
By Shoptail

On Cloudvent Waterproof...
€179.95
On
★★★★★ (114)
By Google

ASICS™ IE: Running Shoes - ASICS™ Official Online Store
Free delivery and free returns on all orders. Official range available here. Shop genderless **running shoes** at ASICS™ online. Extended range. New collections. Online exclusives.

Running Shoes Cross Country Spikes | SportsShoes.com
Shop our range of **Cross Country Shoes** at SportsShoes.com. We're experts in what we do.

View The Best Cross Country Shoes, Below.

1. Salomon Men's Speedcross 4 Trail. [VIEW ON AMAZON](#) ...
2. Saucony Men's Kilkenny XC5. [VIEW ON AMAZON](#) ...
3. New Balance Men's 900v4. [VIEW ON AMAZON](#) ...
4. New Balance Women's 700v5. ...
5. **adidas Performance** Women's XCS. ...
6. Saucony Women's Kilkenny XC5. ...
7. **adidas Performance** Men's XCS. ...
8. Women's Nike Zoom Rival XC.

[More items...](#) • Aug 13, 2020

shoeadviser.com › athletic › best-cross-country-shoes

10 Best Cross Country Running Shoes - Shoe Adviser

About Featured Snippets Feedback

People also ask

What should I wear for cross country running?

Can you use track shoes for cross country?

SEARCH ENGINE RESULTS: SPONSORED/AD AND ORGANIC CONTENT

In the results returned we can distinguish between **organic** and **sponsored** content in the results window – SERP – Search Engine Results Page

- Sponsored specifically refers to ad data, i.e., paid-for-data
- Organic content refers to data found on web pages “for free”
- Sponsored content is listed (ranked) above organic content.

The screenshot shows a Google search results page with the query "running shoes for cross country". The results are divided into two main sections: "Ads" and "Organic".

Ads: This section contains five sponsored listings for running shoes, each with a thumbnail image, product name, price, and a snippet of information. The products listed are:

- Asics Gel Fujitrabuco 7... €89.59 SportsShoes.com ★★★★ (103) By RedBrain
- Brooks Mach 19 Running Shoe... €80.00 Brooks Running ★★★★ (3) By Google
- Nike Air Zoom Pegasus 36 Tr... €105.47 Nike Official Free delivery By PriceSearch...
- New Balance Mens MT590 ... €41.95 MandMDirect.ie By Shoptail
- On Cloudvent waterproof... €179.95 On ★★★★ (114) By Google

Organic Results: Below the ads, there are two organic snippets. The first is from ASICS, advertising their official online store with a link to www.asics.com/. The second is from SportsShoes.com, advertising their range of cross country shoes with a link to www.sportsshoes.com/.

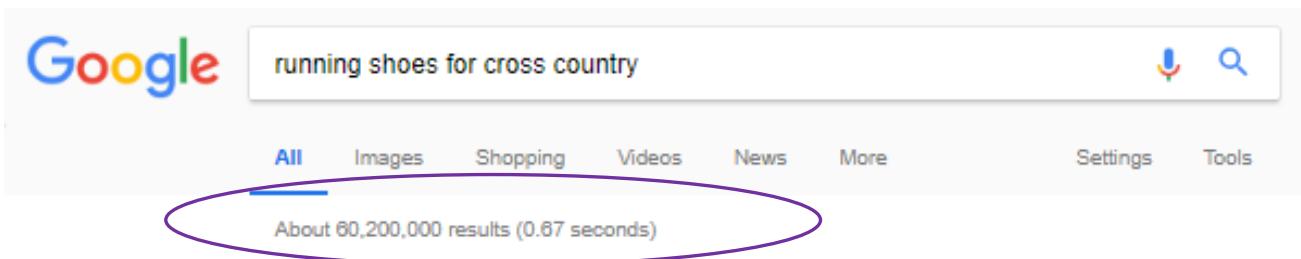
People also ask: At the bottom right, there are two questions with dropdown answers:

- What should I wear for cross country running?
- Can you use track shoes for cross country?

ORGANIC CONTENT



Returning (often millions) of web pages in response to a user query:



(Maybe) looks like: matching user query to web pages
In reality, need something far more **scalable** and **efficient (quick)** than this

SPONSORED CONTENT

Sponsored 

Ad

“Sponsored content” is essentially “[paid-for-ads](#)”

An [additional search](#) occurs independent of the organic search

This search uses a repository ([database](#)) of ad words. If any ad words match the query words and the ad passes some “quality tests” and “wins” at an automated auction then the associated ad is ranked above the web documents returned.

[Examples:](#) Compare searching for “nike runners” and “running shoes review”

SEARCH ENGINE RESULTS: Results are ranked

Ranking involves ordering the results returned in response to a user query

Ranking is based on:

- Business model (e.g., ads first but not all ads)
- Similarity scores (between web pages and query)
- Page rank scores (web links)
- Search Engine Optimisation (SEO)
- Ad word scores
- Personalisation scores: Location, Language, profile settings, past search information, etc. (if used by the search engine)

Cross Country Running Shoes - 7 Things High School Runners ...
<https://www.runnersworld.com/.../7-tips-to-help-high-school-runners-choose-the-right...> ▾
Aug 9, 2018 - As cross-country season begins, it's time for a fresh pair of running shoes. But we know it can be easy to get overwhelmed by the ...

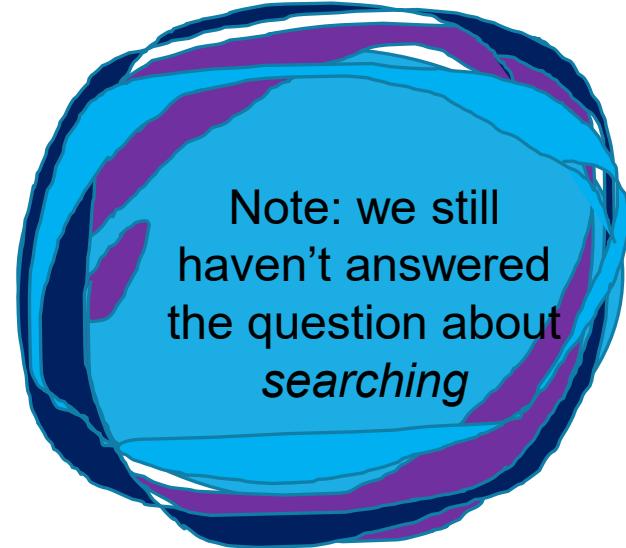
The Best Sneakers for Cross-Country Running | Fitness Magazine
<https://www.fitnessmagazine.com/.../Exercise Equipment/Running Shoes> ▾
The Best Sneakers for Cross-Country Running. Brian Maranan Pineda. New Balance 840. Brian Maranan Pineda. Adidas Supernova Riot. Pearl Izumi Peak XC. These lightweight sneakers are great for fast-paced trail races or training runs. Asics GEL-Trabuco 11 WR. Brooks Cascadia 3. Mizuno Wave Asoend 3. Saucony ProGrid Xodus.

Best Cross Country Shoes Reviewed & Compared in 2018 | RunnerClick
<https://runnerclick.com/10-best-cross-country-shoes-reviewed/> ▾
★★★★★ Rating: 5 - Review by Tess Bercan
Jump to Brooks Running Mach 15 - 10 Best Cross Country Shoes. Salomon Speedcross 4. See more images. ASICS GEL Kayano 25. See more images. Brooks Running Mach 15. See more images. Adidas Supernova Riot M. See more images. La Sportiva Wildcat 2.0 GTX. See more images. Saucony Shay XC4 Flat Shoe. See more images. Pearl Izumi Peak 2. See more images. New ...
Best Cross Country Shoes · Salomon Speedcross 4 · Adidas Supernova Riot M

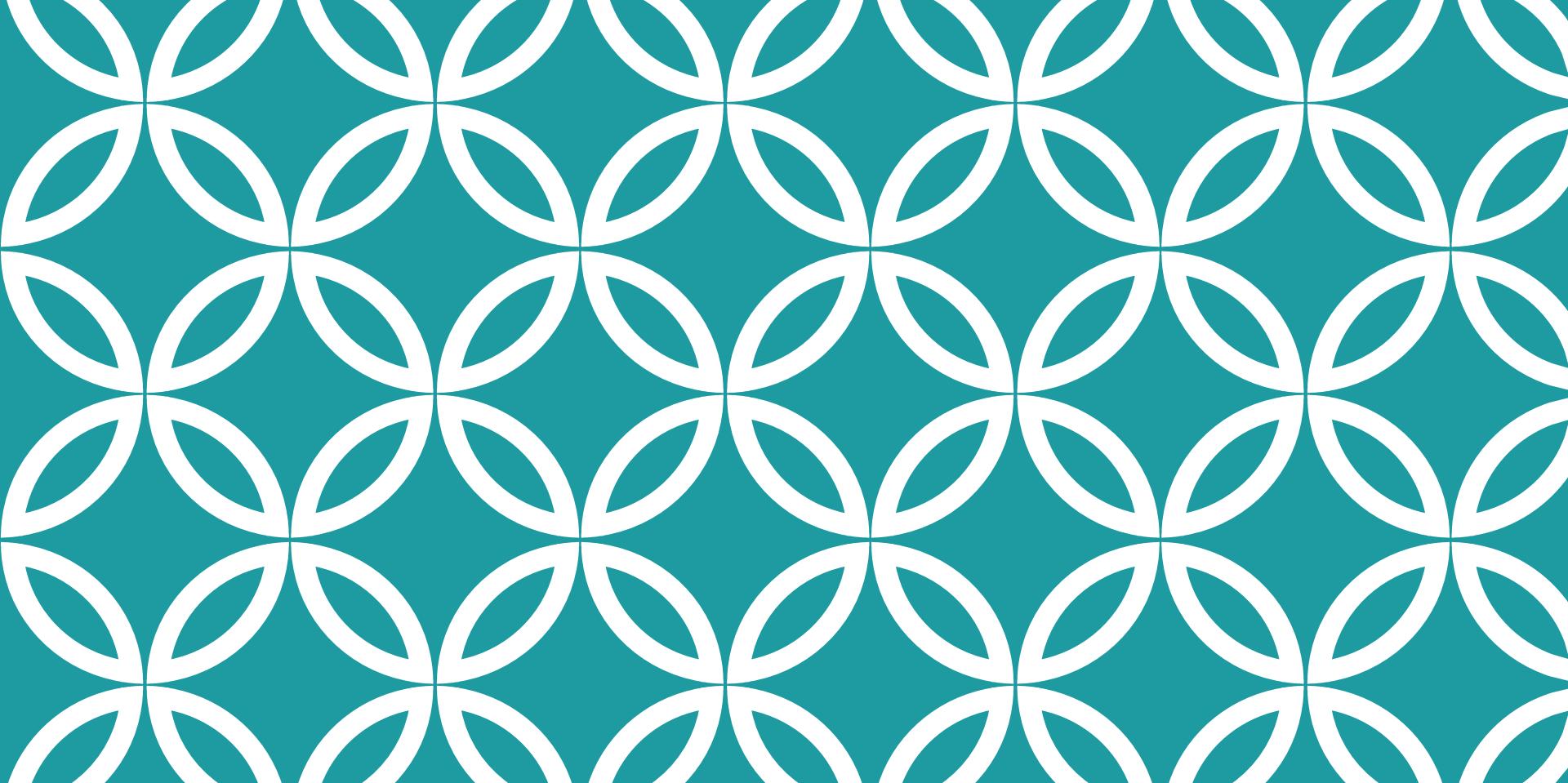
What to Wear For Cross Country Running | Run and Become
<https://www.runandbecome.com/running.../what-to-wear-for-cross-country-running> ▾
Cross Country Spikes. Adidas XCS. Women's Adidas XCS. Brooks Mach 18. Women's Brooks Mach 18. Saucony Havok XC. New Balance XC700 V5. Nike Zoom Rival D 9. Junior Adidas Allroundstar.

Men's Cross Country Shoes - Running Warehouse
<https://www.runningwarehouse.com/catpage-MXC.html> ▾

RECAP:



1. What is web search?
2. What data is used in web search?
3. What do we mean by web search links?
4. What is the difference between structured, unstructured and semi-structured data ... give examples
5. Explain what is meant by sponsored and organic content
6. What does ranking mean? What is web search ranking?



WEB SEARCH:

Web Search Components, Crawling

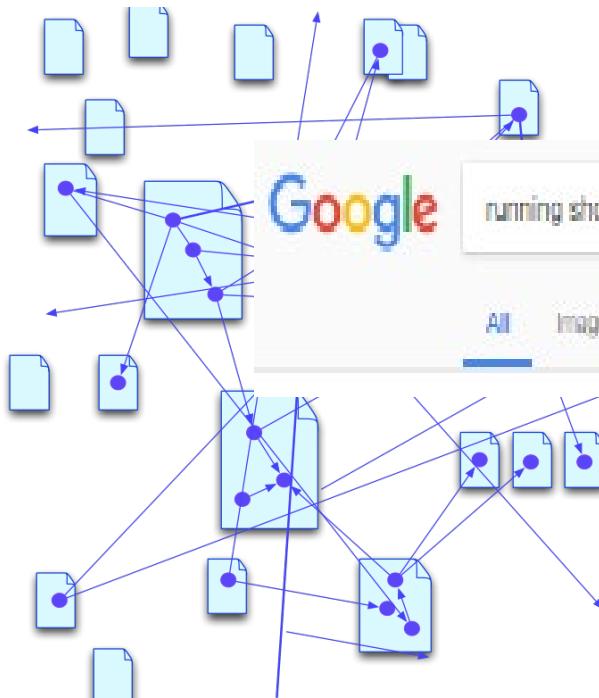
CT102
Information
Systems

SEARCH ENGINE OVERVIEW:

Inputs, processing, outputs

Data → Information

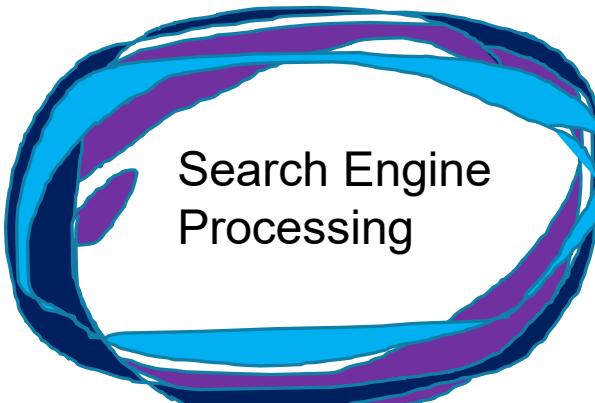
The data?



Input: The query

running shoes for cross country

All Images Shopping News More



The output: The results

Google search results for "running shoes for cross country". The results include:

- ASICS™ IE: Running Shoes - ASICS™ Official Online Store
- www.sportsshoes.com ▾ Running Shoes Cross Country Spikes | SportsShoes.com
- View The Best Cross Country Shoes, Below.
- 1. Salomon Men's Speedcross 4 Trail. VIEW ON AMAZON ...
- 2. Saucony Men's Kilkenny XC5. VIEW ON AMAZON ...
- 3. New Balance Men's 900v4. VIEW ON AMAZON ...
- 4. New Balance Women's 700v5. ...
- 5. adidas Performance Women's XCS. ...
- 6. Saucony Women's Kilkenny XC5. ...
- 7. adidas Performance Men's XCS. ...
- 8. Women's Nike Zoom Rival XC.

More Items... • Aug 13, 2020

shoeadviser.com | athletic | best-cross-country-shoes
10 Best Cross Country Running Shoes - Shoe Adviser

About Featured Snippets Feedback

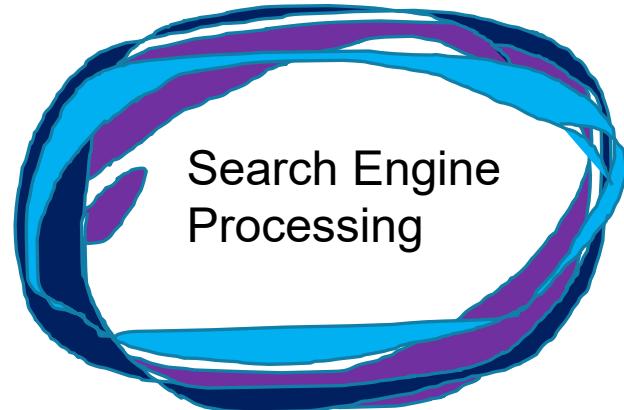
People also ask

What should I wear for cross country running?

Can you use track shoes for cross country?

HOW SEARCH ENGINES WORK

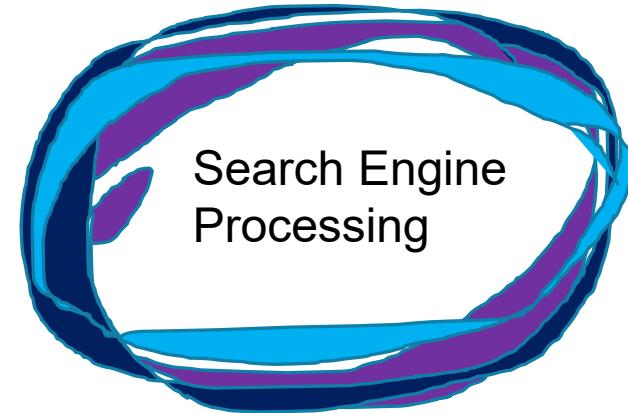
A number of different systems are often part of a single search engine (1 of 2):



- **Organic unstructured content:** Information Retrieval System – matching query terms with terms in the index
- **Personal data:** Personalised System – using personalised data (various forms)
- **Organic HTML links:** Page ranking System – using the existing HTML links between web pages to infer “importance” (no text/images used)

HOW SEARCH ENGINES WORK

A number of different systems are often part of a single search engine (2 of 2):



- **Sponsored data:** Ad System – using matches between keywords in paid ads and the user query keywords and an automated auction
- **Organic Structured Content:** Semantic Web System – using semantically tagged information from linked open data sources (structured data)
- **Displaying results:** Ranking System – a system that takes all the different outputs from the multiple systems and returns a single list of **ranked results** to the user

THE BASIC COMPONENTS OF ORGANIC SEARCH:

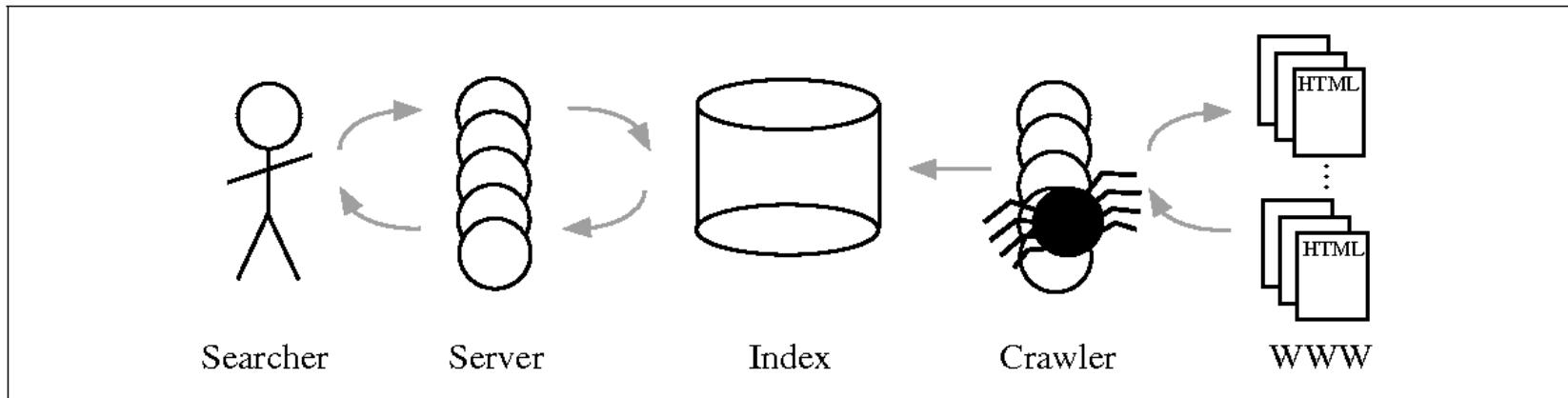


Figure 3.2: WebCrawler's overall architecture. The Crawler retrieves and processes documents from the Web, creating an index that the server uses to answer queries.

From: “Webcrawler: finding what people want” by Pinkerton, Lazowska, Zahorjan, 2000

TYPICAL STAGES IN (ORGANIC) WEB SEARCH (1 OF 2)

Not at search time:

1. Crawl: Navigate the (unstructured) web by *crawling*
2. Parse: *parse* content and extract meaningful terms, links and other information from some portion of current web page
3. Index: Create *indexes* of web pages
4. Rank: Find *page rank* scores of web pages that are indexed

TYPICAL STAGES IN (ORGANIC) WEB SEARCH (2 OF 2)

At Search time, given a user query

5. Use Information Retrieval techniques, such as vector similarity, to find relevant (organic documents) in the index
6. Reorder and display results from 1 based on:
 - a) (already calculated) page rank results based on the importance of a web page
 - b) (* Personalisation step)
7. (** Sponsored content step and personalisation of sponsored content)
8. Search and incorporate structured documents (if applicable)
9. Order and display all results (if they exist) from steps 1-4

PUTTING STEPS 1-6 TOGETHER FOR ORGANIC SEARCH?

THE INTERNET
HOW SEARCH WORKS



https://www.youtube.com/watch?v=LVV_93mBfSU

What other concepts were mentioned in the video?

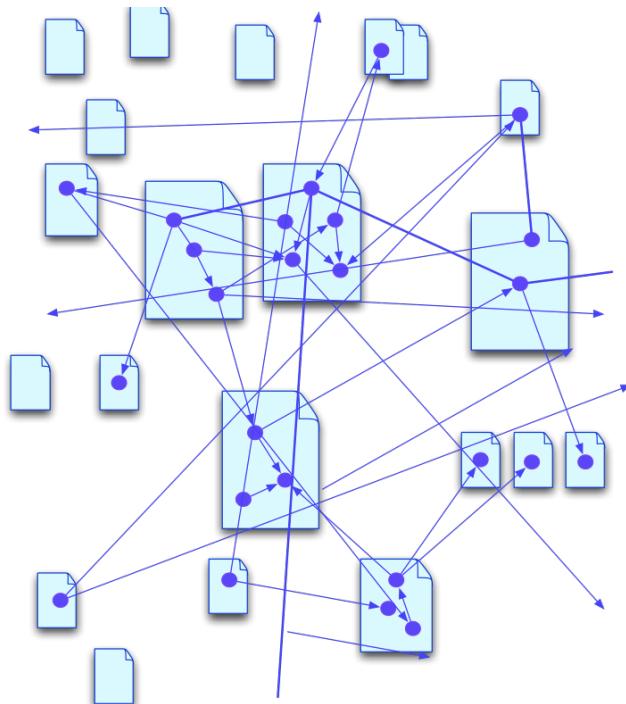
NOW INCORPORATING SPONSORED AND PERSONALISED CONTENT: (steps 1-4 as before)

At Search time, given a user query

5. Use Information Retrieval techniques, such as vector similarity, to find relevant (organic documents) in the index
6. Reorder and display results from 1 based on:
 - a) (already calculated) page rank results based on the importance of a web page
 - b) Personalisation step
7. Sponsored content step and personalisation of sponsored content
8. Search and incorporate structured documents (if applicable)
9. Order and display all results (if they exist) from steps 1-4

Now looking at organic search in more detail

Starting with Step 1: Web Crawling



WEB CRAWLING

- Web crawlers find content (web pages) on the web (independent of any query)
- It is the index that results from crawling websites and parsing the content that is used in live searches
- You can submit your site to search engines but for larger sites (which have many links to them), web crawlers will often find and index a site automatically

OVERVIEW OF CRAWLING:

There exists no central repository or “directory” of all websites so each search engine builds its own

The process of creating and updating this directory is done by “crawlers” (or bots or spiders)

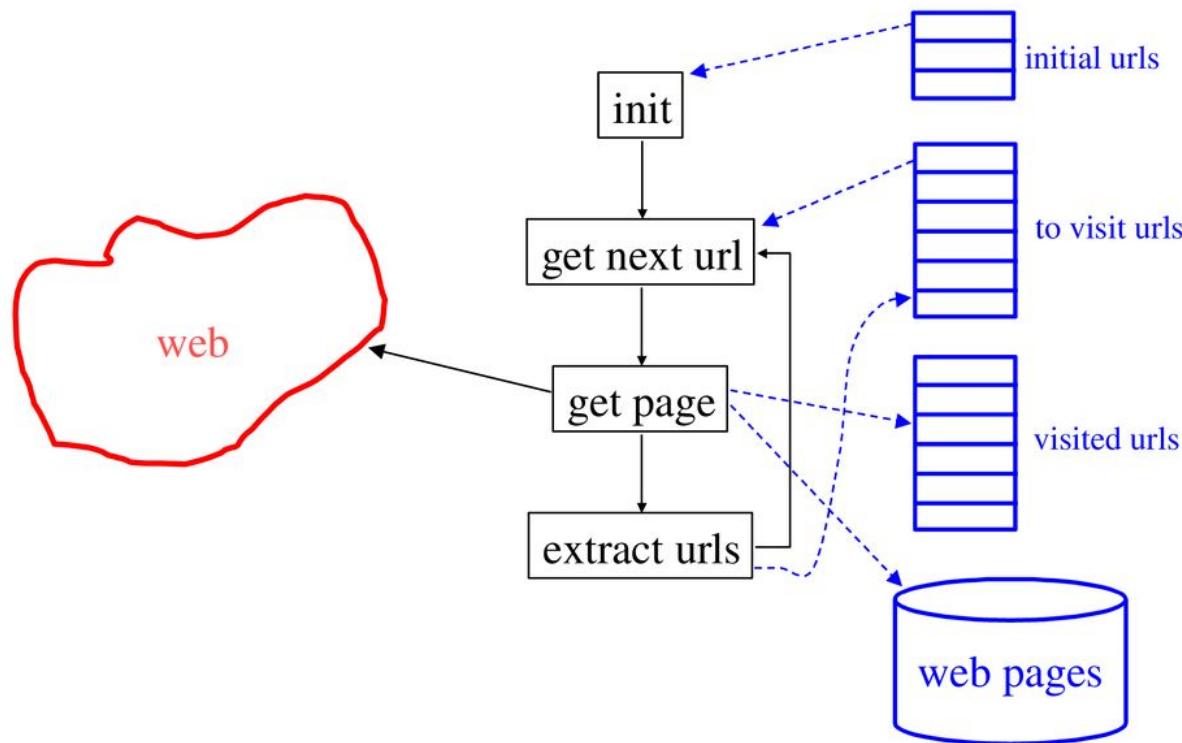
Examples include “Googlebot”, “Slurp” (Yahoo), “bingbot” (Microsoft)

In addition website owners can submit a list of pages (an individual url or a sitemap url) for the crawler to crawl

OVERVIEW OF CRAWLING:

- Starting with a known list of websites, each crawler visits the pages on the list (checking each web pages' robots.txt file) and gets content and finds new pages to add to the list by following a link from a known page to a new page
- Once a page is visited/re-visited (or discovered) the contents (text, images, video) must be scraped or parsed and stored (or updated) in a **searchable index**

WEB CRAWLING OVERVIEW



REP: ROBOTS EXCLUSION PROTOCOL

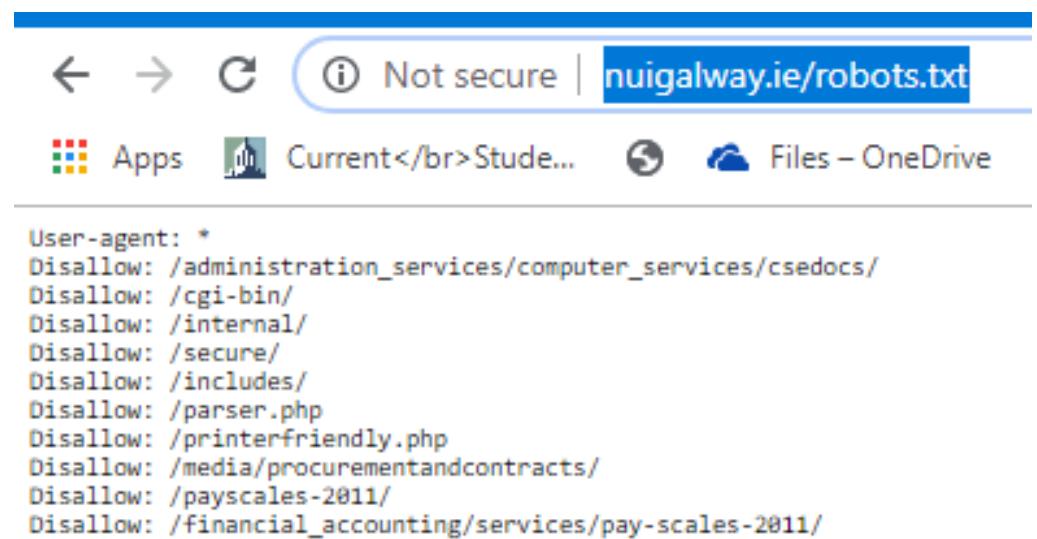
`robots.txt` is a text file webmasters create to instruct crawlers how to crawl pages on their website

The file must be placed in root domain to be found by crawler

When a crawler following REP visits a webpage it will look first for a `robots.txt` file at the root domain.

If one exists, the crawler will follow the instructions in that file before crawling the site and downloading any content

robots.txt



The screenshot shows a browser window with the URL "nuigalway.ie/robots.txt" in the address bar. The page content displays the following text:

```
User-agent: *
Disallow: /administration_services/computer_services/csedocs/
Disallow: /cgi-bin/
Disallow: /internal/
Disallow: /secure/
Disallow: /includes/
Disallow: /parser.php
Disallow: /printerfriendly.php
Disallow: /media/procurementandcontracts/
Disallow: /payscales-2011/
Disallow: /financial_accounting/services/pay-scales-2011/
```

Basic format:

User-agent: [user-agent name]

Disallow: [URL string not to be crawled]

- Might also want to specify a delay for bots that crawl frequently (not supported by all crawlers), e.g.,

Crawl-delay: 10

- And can also indicate where the sitemap is stored (not supported by all crawlers)

EXAMPLES

Blocking all web crawlers from all content

```
User-agent: *
```

```
Disallow: /
```

Blocking one web crawler from a folder:

```
User-agent: Googlebot
```

```
Disallow: /contacts/
```

SAMPLE (*high level*) ALGORITHM FOR CRAWLING WEB (1 of 5)

The crawler begins with one or more URLs that constitute a *seed set* and adds these to the **frontier set**.

The frontier set is a “to do” list of web pages to fetch ... or can view it as an open list of unvisited nodes in the web graph.

The crawler **picks** a URL from the frontier set in some order, e.g. FIFO (first in first out) queue or priority queue

SAMPLE ALGORITHM FOR CRAWLING WEB (2 of 5)

For the URL picked:

- looks up DNS
 - connects to host
 - sends request
 - receives response
 - Based on response ...
wait/retry/redirect/proceed
- * Crawlers need to have **timeouts** so that an unnecessary long amount of time is not spent waiting for a response or reading a web page

SAMPLE ALGORITHM FOR CRAWLING WEB (3 of 5) ... FETCHING WEB PAGES

If proceeding, get robots.txt and using instructions there, proceed to fetch contents and new urls from the web page

The HTTP protocol is used to fetch the web page given the URL picked from the frontier set

Generally do not download an entire web page - only the first portion of a web page – this is seen (and has been proven to be) as *representative enough* of the content of the page

SAMPLE ALGORITHM FOR CRAWLING WEB (4 of 5) ... FETCHING WEB PAGES

The fetched portion of the page is written to a temporary store

The page is **parsed**, to extract both the text, images, video and the links (urls) from the page (each of which points to another web page) and any other information.

Tests are done to see if a web page with the same content has already been seen at another URL or if the page is spam or has been compromised and the index (and/or blacklist) is updated based on these tests

SAMPLE ALGORITHM FOR CRAWLING WEB (5 of 5) ... FETCHING WEB PAGES

If the source is valid and should be stored then:

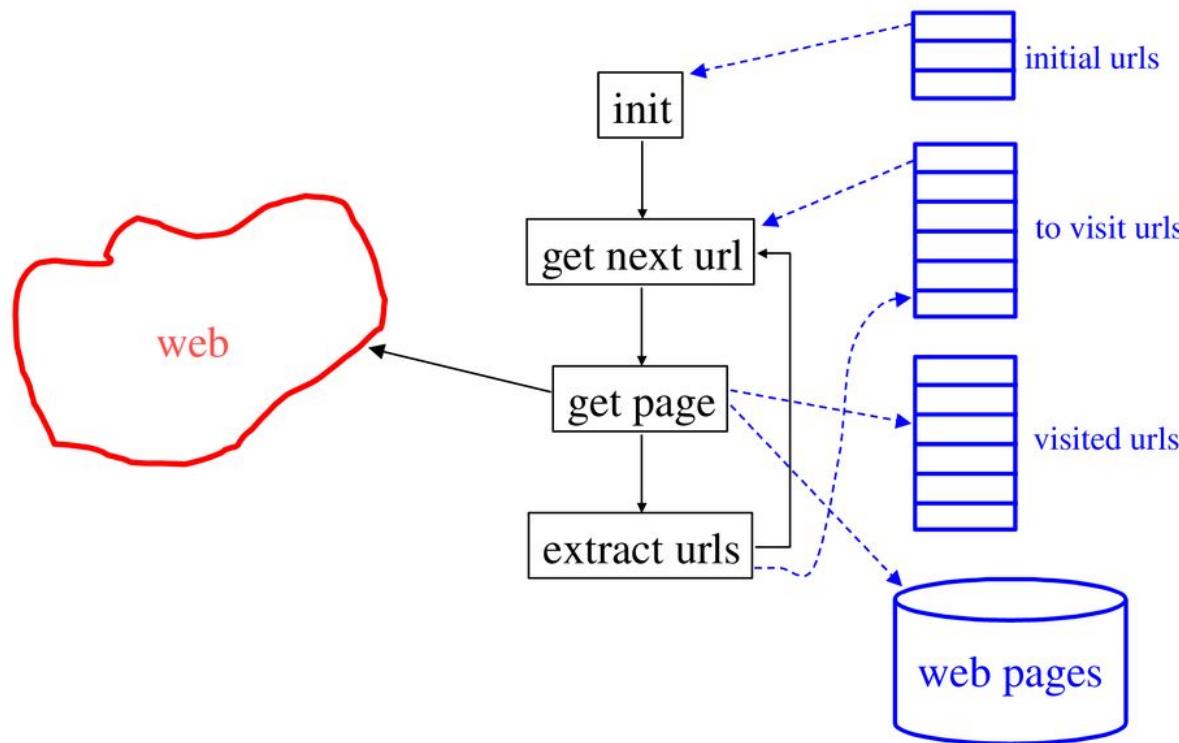
- The extracted text, links and other information are fed to a text **indexer**
- The extracted links (URLs) are then **added** to the *URL frontier set*, if it is not already in the list to be visited later

DISTRIBUTED CRAWLING

In reality crawling is often **distributed**:

- A single URL server sends lists of URLs to a number of crawlers
- Each crawler keeps a number of connections open at once. This is necessary to retrieve web pages at a fast enough pace.
- A major performance stress is DNS lookup so each crawler often maintains its own DNS cache so it does not need to do a DNS lookup before crawling each document.

RECALL: WEB CRAWLING OVERVIEW



DISTRIBUTED CRAWLING

Each of the connections open by a single crawler may be in a number of different states at any one time:

- looking up DNS
- connecting to host
- sending request
- receiving response

Each crawler uses a number of queues to keep track of the status of each connection state

CRAWLER BEHAVIOUR

A crawler requires:

A *selection policy*: to know what order to choose URLs from the frontier list and where to add new URLs

A *re-visit policy*: to indicate when a page already in the index should be revisited to check for additions/deletions and new URLs

A *politeness policy*: to avoid overloading websites by visiting them too often or having too many requests in a short time period

A *parallelization policy*: to coordinate distributed web crawlers

WRITING YOUR OWN CRAWLER?

Once you can send a HTTP request to a URL you will be able to get the content at that URL

Will require further coding to parse the content to extract the meaningful content

**** be careful of being blacklisted**

Many crawlers and indexers exist, e.g.,

<https://www.octoparse.com/>

Google is making its crawler code open source and many open source implementations on github

INDEXING

Indexing organises and stores the data gathered by the crawling stage so that it can be searched in a quick and efficient manner

Typically stores some version of

`<url, term, weight>`

Typically some type of compression is used

Two aspects:

- Weighting of terms to represent their importance
- Storage for fast retrieval (indexing and hashing structures)
(and also reduces the space used)

Google index approx. 60 trillion web pages?

SUMMARY: OWN REFLECTIONS AND QUESTIONS

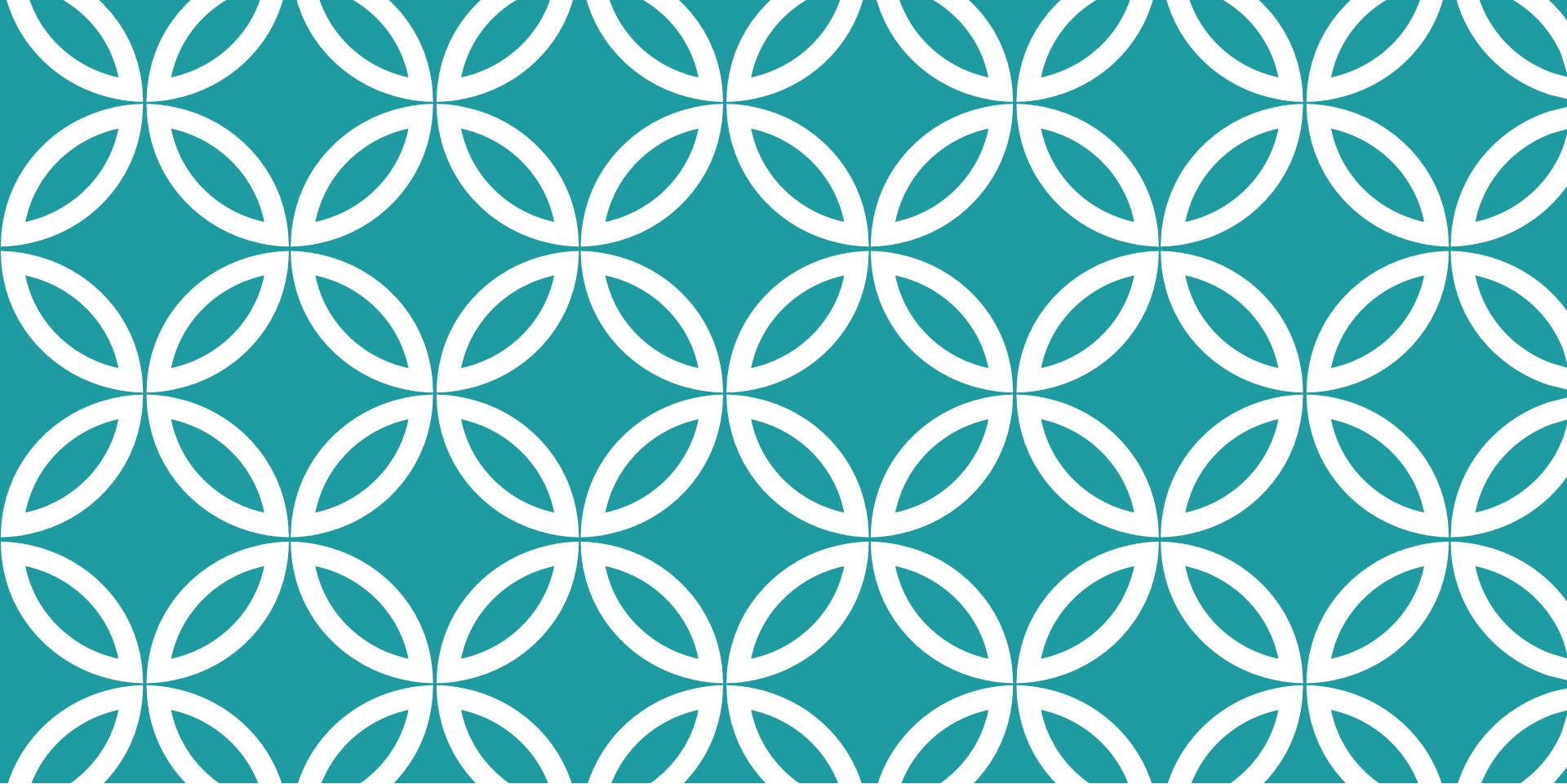
What did you learn?

What are the important things to remember?

What are your questions?

SUMMARY ... after this lecture you should be able to answer the following:

1. What is “organic” web search?
2. How does web search work? / What are the different stages in web search?
3. What is web crawling and why is it used?
4. What are the typical stages in web crawling?
5. What is distributed web crawling?
6. What is a web search index?



WEB SEARCH Indexing:

*Pre-processing and
weighting terms*

CT102
Information
Systems

How to represent/index WWW organic data?

For a web pages (or any document to be searched) need to extract (programmatically) some **abstract representation** to support complex matching (between web page and query) and to speed up querying, i.e. full web page is not searched.

This **abstract representation** is typically created automatically and involves choosing a subset of words from the web page and giving these words certain weights that indicate their importance in describing the web page.

INDEXING OF “ORGANIC” WWW PAGES

An index associates a web page
with one or more terms

A term may be associated with
many web pages

*Automatic indexing begins with no
predefined set of index terms*

These indexes are *dynamic* and
stored on the web search engine
servers in data stores

Index

.....
■ A ■
Alexandria, 29
Anderson, W. C., 49-50
Anna, Lucinda, 9
Antioch, 29
Armentrout, Charles J., 55
Atterbury, John G., 45, 52, 75
Atterbury, John Guest, 14
Austin, W. L., 53
Ayres, Elias, 12-14, 64-65
Bishop, John M., 52
Black, John, 52
Bloomington, 47
Board of Trustees, 15
Boy Hollow, 20-21
Breck, R. L., 15, 49
Brooks, James, 10, 15, 17, 23, 46, 77-78,
103
Brown, Caroline M., 24
Brown, Deacon Jesse J., 21
Brown, Jesse J., 21, 23-24, 26, 46, 53,
95-96
Brown, Sherry Scribner, 31-32, 60, 62
Buren, Martin Van, 7

■ B ■
Baltimore, 23
Bank, D. C., 8, 48
Barksdale, David, 17, 22, 43, 86, 95, 107
Beadle, E. R., 17, 52
Beers, Stephen, 9
Bego, Herman, 36
Bentley, James, 12
Camp Pyoca, 40
Canada, 14
Capernaum, 41
Carille, A. D., 53
Chapel, McCulloch, 25-26
Chicago, 12, 18, 21, 45

INTRODUCING A SAMPLE TEXT ...

Adapted from that available on wikipedia on William Shakespeare

Wish to (programmatically/automatically) find:

- What is the text about? (its meaning)
- Which words help us determine what the meaning is?
- How to automatically extract these words and weight them so they can be added to the index

INTRODUCING A SAMPLE TEXT ...

William Shakespeare (bapt. 26 April 1564 – 23 April 1616) was widely regarded as the greatest writer in the English language and often called England's national poet and the "Bard of Avon". His extant works, including collaborations, consist of some 39 plays,[c] 154 sonnets, two long narrative poems, and a few other verses, some of uncertain authorship. His plays have been translated into every major living language and are performed more often than those of any other playwright.[7]

Shakespeare was born and raised in Stratford-upon-Avon, Warwickshire. At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna and twins Hamnet and Judith. Sometime between 1585 and 1592, he began a successful career in London as an actor, writer, and part-owner of a playing company called the Lord Chamberlain's Men, later known as the King's Men. At age 49 (around 1613), he appears to have retired to Stratford, where he died three years later. Few records of Shakespeare's private life survive; this has stimulated considerable speculation about such matters as his physical appearance, his sexuality, his religious beliefs, and whether the works attributed to him were written by others.[8][9][10] Such theories are often criticised for failing to adequately note that few records survive of most commoners of the period.

Shakespeare produced most of his known works between 1589 and 1613.[11][12][d] His early plays were primarily comedies and histories and are regarded as some of the best work produced in these genres. Until about 1608, he wrote mainly tragedies, among them Hamlet, Othello, King Lear, and Macbeth, all considered to be among the finest works in the English language.[2][3][4] In the last phase of his life, he wrote tragicomedies (also known as romances) and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy in his lifetime. However, in 1623, two fellow actors and friends of Shakespeare's, John Heminges and Henry Condell, published a more definitive text known as the First Folio, a posthumous collected edition of Shakespeare's dramatic works that included all but two of his plays.[13] The volume was prefaced with a poem by Ben Jonson, in which Jonson presciently hails Shakespeare in a now-famous quote as "not of an age, but for all time".[13]

Throughout the 20th and 21st centuries, Shakespeare's works have been continually adapted and rediscovered by new movements in scholarship and performance. His plays remain popular and are studied, performed, and reinterpreted through various cultural and political contexts around the world.

Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tours
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

In other projects
Wikimedia Commons
Wikibooks
Wikisource
Print/export
Create a book

William Shakespeare

From Wikipedia, the free encyclopedia

This article is about the poet and playwright. For other persons of the same name, see William Shakespeare (disambiguation). For other uses of "Shakespeare", see Shakespeare (disambiguation).

William Shakespeare (bapt. 26 April 1564 – 23 April 1616)[2] was an English poet, playwright, and actor, widely regarded as the greatest writer in the English language and the world's greatest dramatist.[2][3][4] He is often called England's national poet and the "Bard of Avon".[5][6] His extant works, including collaborations, consist of some 39 plays,[d] 154 sonnets, two long narrative poems, and a few other verses, some of uncertain authorship. His plays have been translated into every major living language and are performed more often than those of any other playwright.[7]

Shakespeare was born and raised in Stratford-upon-Avon, Warwickshire. At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna and twins Hamnet and Judith. Sometime between 1585 and 1592, he began a successful career in London as an actor, writer, and part-owner of a playing company called the Lord Chamberlain's Men, later known as the King's Men. At age 49 (around 1613), he appears to have retired to Stratford, where he died three years later. Few records of Shakespeare's private life survive; this has stimulated considerable speculation about such matters as his physical appearance, his sexuality, his religious beliefs, and whether the works attributed to him were written by others.[8][9][10] Such theories are often criticised for failing to adequately note that few records survive of most commoners of the period.

Shakespeare produced most of his known works between 1589 and 1613.[11][12][d] His early plays were primarily comedies and histories and are regarded as some of the best work produced in these genres. Until about 1608, he wrote mainly tragedies, among them Hamlet, Othello, King Lear, and Macbeth, all considered to be among the finest works in the English language.[2][3][4] In the last phase of his life, he wrote tragicomedies (also known as romances) and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy in his lifetime. However, in 1623, two fellow actors and friends of Shakespeare's, John Heminges and Henry Condell, published a more definitive text known as the First Folio, a posthumous collected edition of Shakespeare's dramatic works that included all but two of his plays.[13] The volume was prefaced with a poem by Ben Jonson, in which Jonson presciently hails Shakespeare in a now-famous quote as "not of an age, but for all time".[13]



The Chandos portrait (held by the National Portrait Gallery, London)

Born Stratford-upon-Avon, Warwickshire, England
Baptised 26 April 1564
Died 23 April 1616 (aged 52)
Stratford-upon-Avon, Warwickshire, England

is

William Shakespeare

From Wikipedia, the free encyclopedia



This article is about the poet and playwright. For other persons of the same name, see [William Shakespeare \(disambiguation\)](#). For other uses of "Shakespeare", see [Shakespeare \(disambiguation\)](#).

William Shakespeare (bapt. 26 April 1564 – 23 April 1616)^[6] was an English poet, playwright, and actor, widely regarded as the greatest writer in the English language and the world's greatest dramatist.^{[7][8]} He is often called England's national poet and the "Bard of Avon".^[9] His extant works, including collaborations, consist of some 39 plays,^[c] 154 sonnets, two long narrative poems, and a few other verses, some of uncertain authorship. His plays have been translated into every major living language and are performed more often than those of any other playwright.^[7]

Shakespeare was born and raised in Stratford-upon-Avon, Warwickshire. At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna and twins Hamnet and Judith. Sometime between 1585 and 1592, he began a successful career in London as an actor, writer, and part-owner of a playing company called the Lord Chamberlain's Men, later known as the King's Men. At age 49 (around 1613), he appears to have retired to Stratford, where he died three years later. Few records of Shakespeare's private life survive; this has stimulated considerable speculation about such matters as his physical appearance, his sexuality, his religious beliefs, and whether the works attributed to him were written by others.^{[8][9][10]} Such theories are often criticised for failing to adequately note that few records survive of most commoners of the period.

Shakespeare produced most of his known works between 1589 and 1613.^{[11][12]} His early plays were primarily comedies and histories and are regarded as some of the best work produced in these genres. Until about 1608, he wrote mainly tragedies, among them Hamlet, Othello, King Lear, and Macbeth, all considered to be among the finest works in the English language.^{[2][3][4]} In the last phase of his life, he wrote tragicomedies (also known as romances) and collaborated with other playwrights.



WHAT IS THE TEXT ABOUT?

William Shakespeare (bapt. 26 April 1564 – 23 April 1616)[⁶] is widely regarded as the greatest writer in the English language, often called England's national poet and the "Bard of Avon".^{[5][6]} His extant works, including collaborations, consist of some 39 plays,^[c] 154 sonnets, two long narrative poems, and a few other verses, some of uncertain authorship. His plays have been translated into every major living language and are performed more often than those of any other playwright.^[7]

Shakespeare was born and raised in Stratford-upon-Avon, Warwickshire. At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna and twins Hamnet and Judith. Sometime between 1585 and 1592, he began a successful career in London as an actor, writer, and part-owner of a playing company called the Lord Chamberlain's Men, later known as the King's Men. At age 49 (around 1613), he appears to have retired to Stratford, where he died three years later. Few records of Shakespeare's private life survive; this has stimulated considerable speculation about such matters as his physical appearance, his sexuality, his religious beliefs, and whether the works attributed to him were written by others.^{[8][9][10]} Such theories are often criticised for failing to adequately note that few records survive of most commoners of the period.

WHAT WORDS ARE MOST IMPORTANT IN UNDERSTANDING WHAT THE TEXT IS ABOUT?

At the age of 18, he married Anne Hathaway, with whom he had three children: Susanna and twins Hamnet and Judith. Sometime between 1585 and 1592, he began a successful career in London as an actor, writer, and part-owner of a playing company called the Lord Chamberlain's Men, later known as the King's Men. At age 49 (around 1613), he appears to have retired to Stratford, where he died three years later. Few records of Shakespeare's private life survive; this has stimulated considerable speculation about such matters as his physical appearance, his sexuality, his religious beliefs, and whether the works attributed to him were written by others.^{[8][9][10]} Such theories are often criticised for failing to adequately note that few records survive of most commoners of the period.

Shakespeare produced most of his known works between 1589 and 1613.^{[11][12]} His early plays were primarily comedies and histories and are regarded as some of the best work produced in these genres. Until about 1608, he wrote mainly tragedies, among them Hamlet, Othello, King Lear, and Macbeth, all considered to be among the finest works in the English language.^{[2][3][4]} In the last phase of his life, he wrote tragicomedies (also known as romances) and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy in his lifetime. However, in 1623, two fellow actors and friends of Shakespeare's, John Heminges and Henry Condell, published a more definitive text known as the First Folio, a posthumous collected edition of Shakespeare's dramatic works that included all but two of his plays.^[13] The volume was prefaced with a poem by Ben Jonson, in which Jonson presciently hails Shakespeare in a now-famous quote as "not of an age, but for all time".^[13]

Throughout the 20th and 21st centuries, Shakespeare's works have been continually adapted and rediscovered by new movements in scholarship and performance. His plays remain popular and are studied, performed, and reinterpreted through various cultural and political contexts around the world.

Words occurring most frequently?

an	44
and	26
his	13
in	32
of	21

play*	13
shakespeare	9
the	26
wrote/write	4
work	7

Looking at a smaller portion of the paragraph ...

Shakespeare produced most of his known work between 1590 and 1613. Shakespeare's early plays were mainly comedies and histories. After which, Shakespeare wrote mainly tragedies until about 1608, including Hamlet, King Lear, and Macbeth. In his last phase, Shakespeare wrote tragicomedies and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy during his lifetime, and in 1623, two of Shakespeare's former theatrical colleagues were involved in publishing the First Folio, a collected edition of Shakespeare's dramatic works that included all but two of the plays now recognised as Shakespeare's.

WORDS OCCURRING MOST FREQUENTLY?

Shakespeare produced most of his known work between 1590 and 1613. Shakespeare's early plays were mainly comedies and histories. After which, Shakespeare wrote mainly tragedies until about 1608, including Hamlet, King Lear, and Macbeth. In his last phase, Shakespeare wrote tragicomedies and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy during his lifetime, and in 1623, two of Shakespeare's former theatrical colleagues were involved in publishing the First Folio, a collected edition of Shakespeare's dramatic works that included all but two of the plays now recognised as Shakespeare's.

HOW TO DEFINE IMPORTANT WORDS? . . .

in terms of *meaning*

Is each type of word equally important?

- nouns, verbs, articles, etc. For example: “Shakespeare”, “plays”, “of”

Are upper and lower case words different (in terms of meaning)?

- “many” Vs “Many”

Are plural and singular words and different tenses of words **very** different in terms of meaning?

- “play” Vs “plays” Vs “played”

Is Punctuation significant (in terms of meaning)?

- “It’s” Vs “it is”, “He’s” Vs “He is”, “Shakespeare” Vs “Shakespeare’s”

OTHER ASPECTS OF WORDS/LANGUAGES?

How to deal with different words for same meaning?

- “weep” Vs “cry” Vs “lament”

How to deal with a word that has multiple meanings

- “suit”, “bank”, “fly”, “lawn”, “duck”, “pitcher”, “play”

WHAT IS AN “IMPORTANT” WORD?

An important word is one that give us the most information about what the web page is about

That is, the word that tell us about the *meaning of the content of the web page*

HOW TO DEFINE “IMPORTANT” WORDS?

Each unique word important?

- No – just Nouns and Verbs (mostly)

Upper and Lower case words different?

- No - upper and lower case versions of the same word be treated as the same (except for proper nouns)

Plural and Singular/Tenses different words?

- No - should be treated as the same word

Punctuation significant?

- No - should not be considered to give different meanings

How to deal with different words for same meaning?

- Need a thesaurus

How to deal with a word that has multiple meanings

- Need to use the other words surrounding the word to **disambiguate** the word

.... Words become *terms*

In automatic indexing, due to many versions of a word being considered the same, the terminology of **term** is used to encompass all versions of a word.

e.g.,

term = rain

Sample words = rains, raining, rained

Indexing: finding the best terms automatically aka *pre-processing*

For each web page (or fragment) a number of *pre-processing* steps are carried out:

- Case folding: words are changed to lowercase (may be special cases for proper nouns)
- Punctuation is removed (punctuation removal)
- “stop words” are removed (stop word removal)
- “Stemming” is performed

CASE FOLDING:

WORDS ARE CHANGED TO LOWERCASE

words are changed to lowercase

In computing, unless strings are **exactly** the same they will not be considered equal

e.g.,

- ‘Example’ and ‘example’ are not the same
- ‘eXample’ and ‘example’ are not the same

However there is no difference in meaning between the uppercase and lowercase versions.

Therefore, *in general* all strings should be changed to one case – lowercase is the convention (“case folding”)

Exceptions are added for proper nouns

Punctuation is removed

Simple punctuation, such as , . ; - gives little meaning

Other punctuation is a short-hand version of two words, e.g. “she’s”, “they’ll”

Other punctuation is more complex and relates to the word following the punctuation e.g., “shakespeare’s plays”

In general, it is too costly in terms of computation effort to distinguish between different types of punctuation and so it is usually removed and replaced with a space.

N.B. As part of punctuation removal, any “trailing” letters left behind are removed as part of stop word removal (rather than being augmented)

e.g.

she's → she s want she

they'll → they ll want they

DEALING WITH PROPER NOUNS

In English, we know that the first word at start of every sentence begins with a capital letter.

In addition, proper nouns which can occur anywhere in a sentence, have the first letter in capitals, e.g. placenames, people's names, etc. It is often important to treat proper nouns as a special case and not to change them to lowercase.

- Punctuation and where a word occurs in a sentence can be used to distinguish these special cases.

Note that abbreviations (e.g., EU, USA, NPHE, etc.) will generally all be in uppercase and may also remain in uppercase.

- These can be distinguished by the fact that they are all uppercase or that they contain “non-standard” punctuation occurrences, e.g., U.S.A.

Task: Carry out the 1st two steps for 1st paragraph of Shakespeare example with no special case for proper nouns

Shakespeare produced most of his known work between 1590 and 1613. Shakespeare's early plays were mainly comedies and histories. After which, Shakespeare wrote mainly tragedies until about 1608, including Hamlet, King Lear, and Macbeth. In his last phase, Shakespeare wrote tragicomedies and collaborated with other playwrights.

shakespeare produced most of his known work between 1590 and 1613 shakespeare early plays were mainly comedies and histories after which shakespeare wrote mainly tragedies until about 1608 including hamlet king lear and macbeth in his last phase shakespeare wrote tragicomedies and collaborated with other playwrights

Stop word removal

Stop words are words that do not provide any extra information about the meaning of a document

Stop words are very common (frequently occur) in a document and often have a small number of letters

Examples are language specific. In English: [the](#), [a](#), and

Stop words are removed to save storage space and to speed up searches

The tendency now is to have a quite small list of stop words

No common set is used – depends on domain – different stop words would be used for Twitter data than for web page data

SAMPLE ENGLISH STOP WORD LIST (stopwords1.txt)

a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because, been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your

from: <http://www.textfixer.com/resources/common-english-words.txt>

LIST POSSIBLY USED BY GOOGLE (stopwords2.txt)

- a
- for
- often
- very
- about
- from
- on
- via
- above
- how
- or
- was
- an
- i
- than
- what
- and
- if
- that
- when
- are
- in
- the
- where
- as
- is
- these
- whether
- at
- it
- they
- who
- be
- not
- this
- will
- by
- of
- to
- with

KEVIN BOUGE STOP WORD LIST

A much longer list of stop words and available in many languages - Arabic, Armenian, Brazilian, Bulgarian, Chinese, Czech, Danish, Dutch, English, Farsi, Finnish, French, German, Greek, Hindi, Hungarian, Indonesian, Italian, Japanese, Latvian, Norwegian, Polish, Portuguese, Romanian, Russian, Spanish, Swedish, Turkish.

<https://sites.google.com/site/kevinbouge/stopwords-lists>

a
a's
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
ain't
all
allow
allows
almost
alone
along
already
also
although
always
am
among
amongst
an
and
another
any
anybody
anyhow
anyone
anything
anyway

APPROACH FOR STOP WORD REMOVAL:

- When a document is initially processed, each word is checked against a stop word list. If the word is not on list it is output to new file; if word is found then it is not output
- Each query should also be processed against a stop list
- High level algorithmic steps:

while not EOF do:

 read in line

 for each word in line:

 if word **not** in stop list:

 write word to new file

IMPROVED APPROACH:

Before the stop word list is checked, find the length of each word (`len(word)`)

Remove all words of length 1 and 2

This is easy to implement and means that a much shorter stop word list can be used if words of length 1, 2 (and maybe 3) do not have to be checked against the stop word list.

Stop word removal for portion of Shakespeare example using `stopwords1.txt`

shakespeare produced most of his known work between 1590 and 1613 shakespeare early plays were mainly comedies and histories after which shakespeare wrote mainly tragedies until about 1608 including hamlet king lear and macbeth in his last phase shakespeare wrote tragicomedies and collaborated with other playwrights

shakespeare produced known work between 1590 1613
shakespeare early plays mainly comedies histories
shakespeare wrote mainly tragedies until 1608 including
hamlet king lear macbeth last phase shakespeare wrote
tragicomedies collaborated playwrights

NOTE: Reduction in number of terms

Original paragraph has 46 words

After stop word removal, there are 31 words left

shakespeare produced most of his known work between 1590 and 1613
shakespeare early plays were mainly comedies and histories after which
shakespeare wrote mainly tragedies until about 1608 including hamlet king
lear and macbeth in his last phase shakespeare wrote tragicomedies and
collaborated with other playwrights

shakespeare cwork between 1590 1613 shakespeare early plays mainly
comedies histories shakespeare wrote mainly tragedies until 1608 including
hamlet king lear macbeth last phase shakespeare wrote tragicomedies
collaborated playwrights

STEMMING

- Stemming tries to find the “stem” of each word.
- A stem represents variant forms of a word which share a common meaning.
- The approach used is language specific.
- Assuming words are written left to right (as in English), then the stem is on the left and letters are often removed on the right.
- As part of stemming, zero or more suffixes may also be added on the right.

Here is a sample of vocabulary, with the stemmed forms that will be generated with the algorithm.

word	stem	word	stem
consign	consign	knack	knack
consigned	consign	knackeries	knackeri
consigning	consign	knacks	knack
consignment	consign	knag	knag
consist	consist	knave	knave
consisted	consist	knaves	knave
consistency	consist	knavish	knavish
consistent	consist	kneaded	knead
consistently	consist	kneading	knead
consisting	consist	knee	knee
consists	consist	kneel	kneel
consolation	consol	kneeled	kneel
consolations	consol	kneeling	kneel
consolatory	consolatori	kneels	kneel
console	consol	knees	knee
consoled	consol	knell	knell
consoles	consol	knelt	knelt
consolidate	consolid	knew	knew
consolidated	consolid	knick	knick
consolidating	consolid	knif	knif
consoling	consol	knife	knife
consolingly	consol	knight	knight
consols	consol	knightly	knight
consonant	conson	knights	knight
consort	consort	knit	knit
consorted	consort	knits	knit
consorting	consort	knitted	knit

FOR EXAMPLE: Stem of these terms?

connected

connection

connecting

connections

connect

computing

computers

computed

computations

compute

comput

worried

worries

worrying

worri

HOW DOES STEMMING WORK?

- Consists of many set of rules that are checked in a certain order
- Terms are usually stemmed as part of pre-processing (after stop word removal) to avoid stemming stop words
- The commonly-used stemming algorithms (for English) are called **Porter's Stemming Algorithm**, **Snowball Stemmer (Porter 2)** and **Lancaster Stemming algorithm**
- Stemming does not work for all languages (e.g. Chinese)

SAMPLE RULES (1 OF 2)

if (word ends in ‘ies’) :

remove ‘ies’

add ‘y’

e.g., **pastries** → **pastry**

ponies → **pony**

berries → **berry**

SAMPLE RULES (2 OF 2)

If (word ends 'es' but not in 'oes'):

remove 's'

e.g.,

files → file

ceases → cease

potatoes →

banjoes →

TRY IT ONLINE ...

Interactive version:

Snowball (and others): <http://text-processing.com/demo/nstem/>

People mostly use existing implementations and do not re-code it (due to complexity of rules):

See:

<http://tartarus.org/~martin/PorterStemmer/>

<http://snowball.tartarus.org/algorithms/english/stemmer.html>

Try: Stemming for portion of Shakespeare example with Snowball English stemmer from <http://text-processing.com/demo/stem/>

shakespeare produced known work between 1590 1613
shakespeare early plays mainly comedies histories
shakespeare wrote mainly tragedies until 1608 including
hamlet king lear macbeth last phase shakespeare wrote
tragicomedies collaborated playwrights

shakespear produc known work between 1590 1613
shakespear earli play main comedi histori shakespear
wrote main tragedi until 1608 includ hamlet king lear
macbeth last phase shakespeare wrote tragicomedi
collabor playwright

LEMMATISATION

A lemma is a base form (core) of a word and it is what we look up in a dictionary

Lemmatisation is the conversion of a word to its lemma

e.g.,

walking → walk

walked → walk

goose → goose (stem: goos)

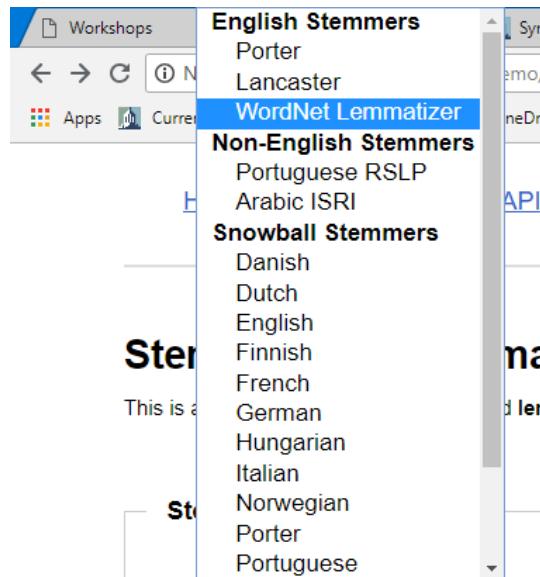
geese → goose (stem: gees)

Finding the lemma of a word is much harder (automatically) than finding a stem

TRY IT ONLINE ...

This interactive version has English Lemmatisation also:

<http://text-processing.com/demo/nstem/>



SHAKESPEARE EXAMPLE AGAIN:

shakespeare produced known work between 1590 1613
shakespeare early plays mainly comedies histories
shakespeare wrote mainly tragedies until 1608 including
hamlet king lear macbeth last phase shakespeare wrote
tragicomedies collaborated playwrights

WordNet Lemmatizer:

shakespeare produced known work between 1590 1613
shakespeare early play mainly comedy history
shakespeare wrote mainly tragedy until 1608 including
hamlet king lear macbeth last phase shakespeare wrote
tragedy collaborated playwright

COMPARING RESULTS:

WordNet Lemmatizer:

shakespeare produced known work between 1590 1613
shakespeare early play mainly comedy history
shakespeare wrote mainly tragedy until 1608 including
hamlet king lear macbeth last phase shakespeare wrote
tragicomedy collaborated playwright

Snowball English stemmer:

shakespear produc known work between 1590 1613
shakespear earli play main comedи histori shakespear wrote
main tragedи until 1608 includ hamlet king lear macbeth last
phase shakespear wrote tragicомеди collabor playwright

THESAURUS

Synonyms are different words with identical or very similar meanings

Often important to identify terms which have synonyms

Examples:

- cry/weep/lament
- ill/sick
- thesis/dissertation
- holiday/vacation
- mail/post
- student/pupil

IMPLEMENTATION

Two approaches to include synonyms where a thesaurus can be used:

- To replace each term in a document with its variants (based on the thesaurus)
- To broaden a query by including variants of terms in the query (much more efficient approach)

Online at:

<http://thesaurus.com/>

Looking at all these pre-processing steps for following two Shakespeare paragraphs: *(Note: 97 words)*

Shakespeare produced most of his known work between 1590 and 1613. Shakespeare's early plays were mainly comedies and histories. After which, Shakespeare wrote mainly tragedies until about 1608, including Hamlet, King Lear, and Macbeth. In his last phase, Shakespeare wrote tragicomedies and collaborated with other playwrights.

Many of Shakespeare's plays were published in editions of varying quality and accuracy during his lifetime, and in 1623, two of Shakespeare's former theatrical colleagues were involved in publishing the First Folio, a collected edition of Shakespeare's dramatic works that included all but two of the plays now recognised as Shakespeare's.

Looking at all these pre-processing steps for the two Shakespeare paragraphs: (stopwords2.txt & Porter Stemmer) *(Note: 71 terms)*

shakespear produc most known work between
1590 1613 shakespear earli play were mainli
comedi histori after which shakespear wrote
mainli tragedi until 1608 includ hamlet king lear
macbeth last phase shakespear wrote
tragicomedi collabor other playwright mani
shakespear play were publish edit vari qualiti
accuraci dure lifetime 1623 shakespear former
theatric colleagu were involv publish first folio
collect edit shakespear dramat work includ play
recognis shakespear

TERMS THAT OCCUR MORE THAN ONCE:

shakespear	8
play	3
were	3
edit	2
hi	2
include	2
mainli	2
publish	2
two	2
work	2
write	2

TERMS THAT OCCUR ONCE

... also important!

1590	collect	hamlet	macbeth	theatric
1608	comedi	histori	mani	tragedi
1613	dramat	involv	now	tragicomedi
1623	dure	king	phase	until
accuraci	earli	known	playwright	vari
between	first	last	produc	
collabor	folio	lear	qualiti	
colleagu	former	lifetim	recognis	

CLASS WORK ... QUESTION

For each sentence given show how a pre-processing stage, involving **case change**, **punctuation removal**, **stop word removal** and **stemming**, produces a new representation of each sentence.

Indicate clearly the approaches you are using, listing the stop words you are using and the approach and the general type of stemming rules used.

* You may use an online stemmer (use Snowball) and stopwords2.txt and do not have any special rules for Proper Nouns.

SENTENCES...

3sentences.txt on blackboard

Consider the following three short sentences, s1, s2 and s3, and their contents:

s1: Python is a very powerful programming language.

s2: Python is often compared to the programming languages Perl, Ruby, Scheme and Java.

s3: Python, Perl, Ruby, Scheme, Java- what's the difference and is Python the best?

stopwords2.txt

- a
- for
- often
- very
- about
- from
- on
- via
- above
- how
- or
- was
- an
- i
- than
- what
- and
- if
- that
- when
- are
- in
- the
- where
- as
- is
- these
- whether
- at
- it
- they
- who
- be
- not
- this
- will
- by
- of
- to
- with

PRE-PROCESSING SUMMARY

Indexing automatically scans the web page downloaded by the crawlers for the most important words and converts these to terms following a sequence of steps involving:

- case folding/change
- punctuation removal
- stop word removal
- stemming

These words are then weighted (next topic) and stored as the *representation of the web page*

CALCULATING THE WEIGHTS OF TERMS

The abstraction is: the *meaning* of a document is represented using terms (derived from words in the document) and a weight for each term where:

- A weight is a real number
- The higher the weight the more important the term is in describing the *meaning* of the document
- One approach to calculate the weight is called: tf-idf:
- **Term Frequency - Inverse Document Frequency**

(tf) term frequency

(idf) inverse document frequency

- **tf:** If a term occurs very often in a document it is an important term describing the document (term frequency), e.g. the term **shakespeare** was the most frequent term in the sample text
- **df:** However, if the term occurs often across all documents which are being searched* then it is not very useful at distinguishing one document from another (document frequency for term is high), e.g. if term **shakespeare** occurs frequently in all documents

* In the index

NOTE:

- Stop words will have a high tf and df
- The weighting is performed on the terms remaining **after** stop word removal and stemming
- Thus if a term is not removed as a stop word, but it occurs frequently in most documents, it will get a low weight and not be considered important in determining the meaning of a web page/document

Calculating term frequency (tf)

To not penalise short documents, **normalise** (compare like with like) by dividing the raw count of the number of times the term occurs by the number of total terms in a document:

Term frequency =

Number times a term t occurs in document divided by the number of terms in the document

This ensures longer documents do not get an “unfair” advantage

Does this make sense? Why?

ASIDE (If it's not making sense):

- For a term t the term frequency can be a raw count of the number of times the term occurs in a document
- However, this is not ideal as a term is likely to occur more often in longer documents, thus longer documents would have an unfair advantage over shorter documents
- Thus it is the ratio of a terms occurrence we would like, not the raw count

EXAMPLE

Given the following information for the term “*shakespeare*” in 3 documents of different lengths find the term frequency:

Doc ID	Frequency	Number of terms in doc	$tf = ?$
d1	10	20	
d2	10	200	
d3	100	2000	

Calculating the inverse document frequency (idf)

For a term t and N documents with t occurring in df_t documents the inverse document frequency of t is defined as:

$$idf_t = \log_{10} \left(1 + \frac{N}{df_t} \right)$$

The **idf** of a rare term should be high, whereas the idf of a frequent term should be low.

To prevent multiplication by 0 the 1 is added

NOTE: LOGS

A **logarithm** is the power to which a number must be raised in order to get some other number

e.g.,

If $\log_{10}x = y$ then $10^y = x$

$\log_{10}100 = ?$

On a calculator **log** is usually \log_{10}

Scientific calculator online:

<https://www.calculator.net/scientific-calculator.html>

EXAMPLE FOR TERM *shakespeare*

with $N = 100$ and $df = 40$

doc	tf	idf (with \log_{10})	tf*idf
d1	0.5	\log_{10} $(1+100/40) =$ 0.544	0.272
d2	0.05	0.544	.0272
d3	0.05	0.544	.0272

Therefore, tf-idf for a term t :

- has the highest weight when t occurs many times within a small number of documents (thus lending **high discriminating power** to those documents)
- has a lower weight when the term occurs fewer times in a document, or occurs in many documents
- has the lowest weight when the term occurs in nearly all documents (i.e. stop words)

You try it ...

Compute the tf-idf weights for the words:

- **sql, databases, programming, computer**

with corresponding terms:

sql, database, program, comput

for each of 3 documents using the following information:

- * Use precision to 4 decimal places

Frequency of terms in docs

	d1 (length 90)	d2 (length 100)	d3 (length 50)
sql	12	4	7
database	3	13	0
program	0	13	2
comput	6	0	3

Frequency of terms across 250 documents

<i>Term</i>	<i>Frequency in Collection (df)</i>
sql	81
database	67
program	192
comput	240

Fill in the tf-idf weights

	sql	database	program	comput
d1				
d2				
d3				

*AFTER THIS STAGE
we can easily access:*

For each web page ID (url) many:
 $\langle \text{term}, \text{ weight} \rangle$

For each term in the collection (term) many:
 $\langle \text{doc}, \text{ weight} \rangle$

So with this information **df** can be calculated (and stored)
for any term

These are generally stored in a complex structure
to aid *fast searching and matching* (*with 0s not
stored generally*)

	sql	database	program	comput
d1							
d2							
d3							
...							
....							
....							
df							

CLASS WORK

<https://web2.0calc.com/>

S1: python very power program language

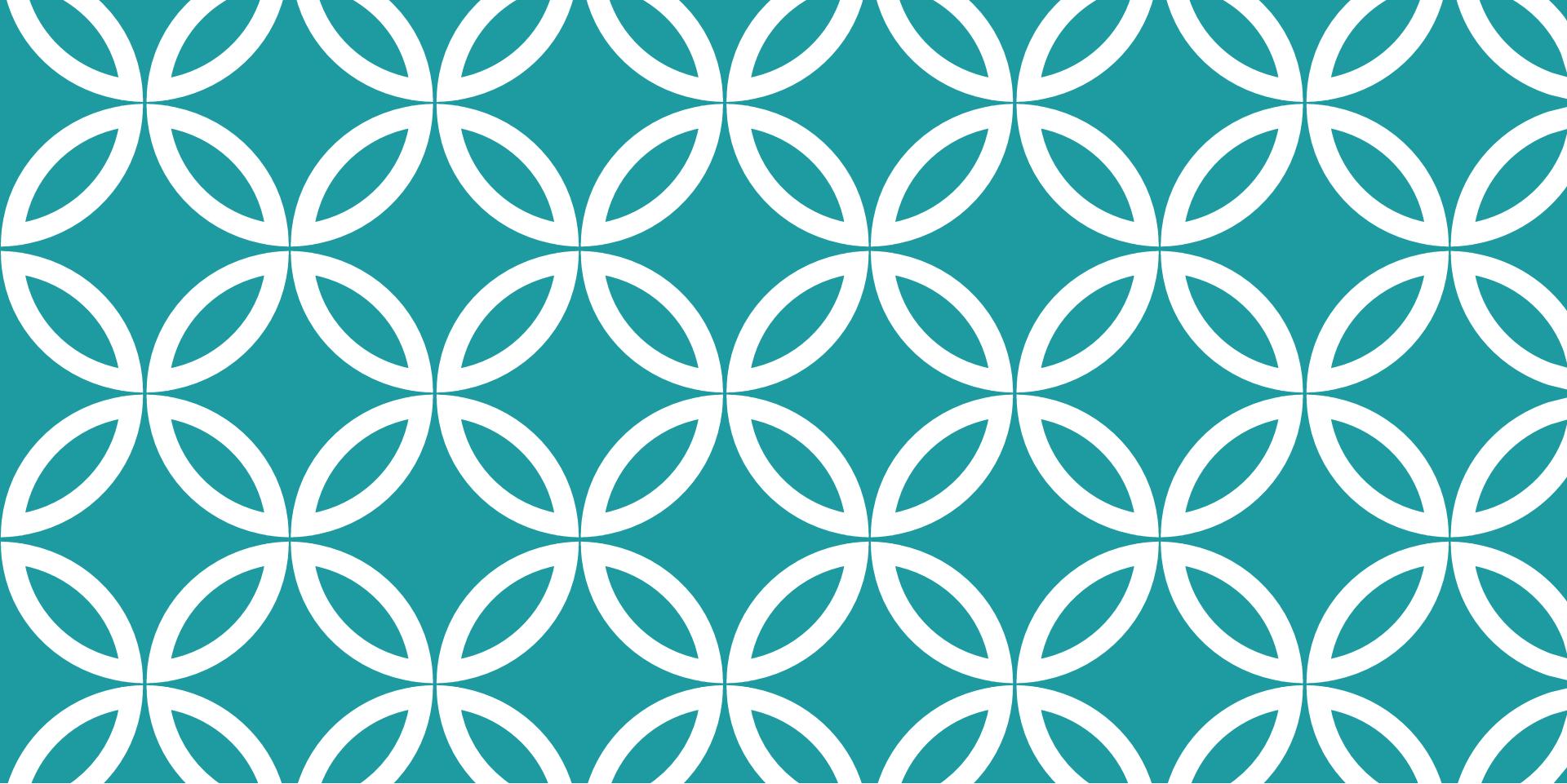
S2: python often compar program language perl
rubi scheme java

S3: python perl rubi scheme java diff python best

1. Find the representation (sentences) after the pre-processing steps of case folding, punctuation removal, stop word removal and stemming have been formed
2. Calculate tf*idf of all the terms remaining in s1 after pre-processing. You can assume that the full document collection is the 3 sentences (to calculate df))

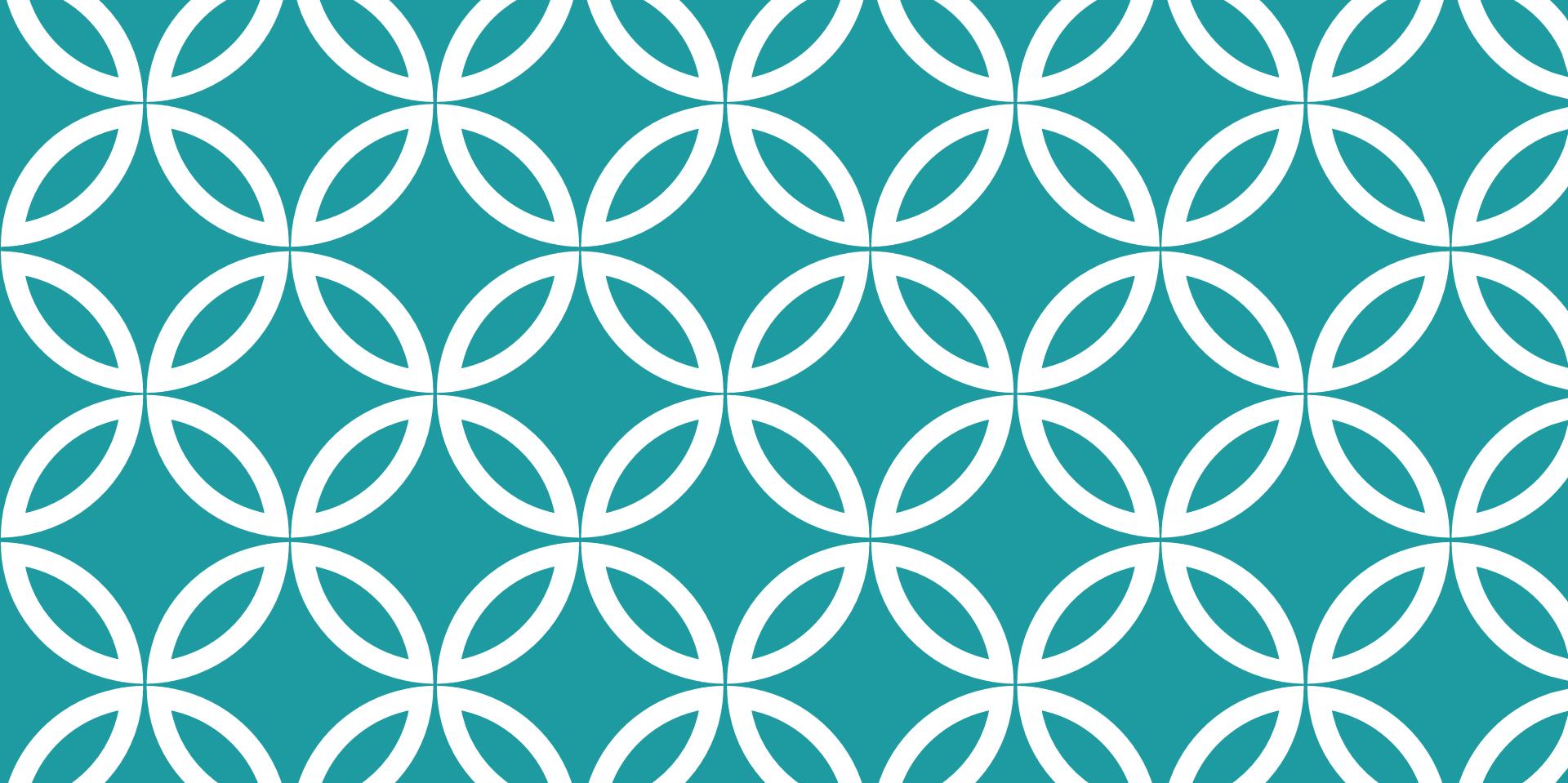
SUMMARY

- A web search index is built based on **term weights** which are calculated after pre-processing steps have been performed
- A commonly used weighting scheme is **tf-idf** (and variations)
 - For **tf** we must know the raw count of the term (frequency) and the total number of words in the document
 - For **idf** we must know the number of documents in the collection and the count of how many of these contain the term



WEB SEARCH: Finding Similarity Page Rank Algorithm

CT102
Information
Systems



FINDING SIMILARITY

(Between web pages and a query)

CT102

EXAMPLE

Portion of 5 documents after case folding and a query

doc1: ... news about republican candidates ...

doc2: ... news about organic food campaign ...

doc3: ... news of republican presidential campaign

doc4: ... news about the presidential campaign and the republican
presidential candidate ...

doc5: news of organic food campaign where the campaign ...

Query: news about republican presidential campaign

a
about
above
an
and
are
as
at
be
by
for
from
how
i
if
in
is
it
not
of
often
on
or
than
that
the
these
they
this
to
very
via
was
what
when
where
whether
who
will
with

LET'S REMOVE STOP WORDS

(docs and query - using `stopword2.txt`)

doc1: ... news ~~about~~ republican candidates ...

doc2: ... news ~~about~~ organic food campaign ...

doc3: ... news ~~of~~ republican presidential campaign

doc4: ... news ~~about the~~ presidential campaign ~~and the~~
republican presidential candidate ...

doc5: news ~~of~~ organic food campaign ~~where the~~ campaign ...

Query: news ~~about~~ republican presidential campaign

NOW STEM (docs and query) using snowball stemmer

<http://text-processing.com/demo/stem/>

doc1: ... news republican candid ...

doc2: ... news organ food campaign ...

doc3: ... news republican presidenti campaign

doc4: ... news presidenti campaign republican presidenti candid ...

doc5: ... news organ food campaign... campaign ...

Query: news republican presidenti campaign

Given the following tf*idf weights were calculated for the same terms for each document

	news	republican	candid	organ	campaign	presidenti	food	sim
doc1	0.2	0.3	0.1					
doc2	0.01			0.4	0.3		0.3	
doc3	0.15	0.35			0.4	0.35		
doc4	0.25	0.32	0.15		0.33	0.4		
doc5	0.1			0.43	0.3		0.5	
query	1	1			1	1		

Assume that query terms have weight of 1

VECTOR SPACE MODEL

Main abstractions:

- For all documents, each document D is represented as a **vector of real-valued numbers** where each number corresponds to the weights of a term in the document
- Queries are also viewed as vectors
- Each position in the vector corresponds to a term from the document collection
 - Therefore, the length of the vector (number of weights/vector dimension) is the number of terms in a document collection called **the vocabulary**

VECTOR SPACE COMPARISON

- A query is also represented as a vector where each term in the query can be assigned a weight of 1.0
- Comparison is done by finding the similarity between document vectors and the query vector
- E.g., in previous example:

$$\overrightarrow{query} = < 1.0 \ 1.0, 0.0, 0.0, 1.0, 1.0, 0.0 >$$

	news	republican	candid	organ	campaign	presidenti	food
doc1	.2	.3	.1	0	0	0	0
doc2	.01			.4	.3	0	.3
doc3	.15	.35			.4	.35	
doc4	.25	.32	.15		.33	.4	
doc5	.1			.43	.3		.5
query	1	1			1	1	

$$\overrightarrow{doc1} = < 0.2, 0.3, 0.1, \textcolor{red}{0.0}, \textcolor{red}{0.0}, \textcolor{red}{0.0}, \textcolor{red}{0.0} >$$

$$\overrightarrow{doc2} = < 0.1, 0.0, 0.0, 0.4, 0.3, 0.0, 0.3 >$$

$$\overrightarrow{doc3} = < 0.15, 0.35, 0.0, 0.0, 0.4, 0.35, 0.0 >$$

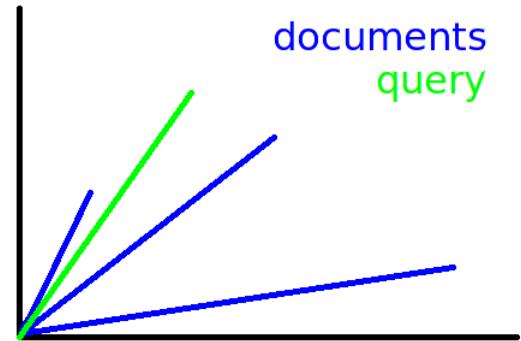
$$\overrightarrow{doc4} = < 0.25, 0.32, 0.15, 0.0, 0.33, 0.4, 0.0 >$$

$$\overrightarrow{doc5} = < 0.1, 0.0, 0.0, 0.43, 0.3, 0.0, 0.5 >$$

$$\overrightarrow{query} = < 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0 >$$

From google images (2D vectors)

Similarity between a document vector d and a query vector q :



The idea is to “measure” the angle or distance between the vectors d and q which represent the document d and query q .

This is done using the **Euclidean dot product** of the two vectors

If the vectors are close (i.e. similar) the distance between them is small and the result is close to 1.

If the vectors are far apart (i.e. dissimilar), the distance between them is large and the result is close to 0.

Similarity between vectors d and q: dot product definition

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n x_i^2} * \sqrt{\sum_1^n q_i^2}}$$

For n terms where:

x_i is weight for i^{th} term in d

q_i is weight for i^{th} term in q

Note that the denominator normalises (by the vector norm or vector magnitude) so that the number of terms in a document is considered. For the denominator we need to consider all terms in the document but for the numerator we only need to consider non-zero weights

PREVIOUS EXAMPLE

	news	republican	candid	organ	campaign	presidenti	food	sim
doc1	.2	.3	.1					
doc2	.01			.4	.3		.3	
doc3	.15	.35			.4	.35		
doc4	.25	.32	.15		.33	.4		
doc5	.1			.43	.3		.5	
query	1	1			1	1		

Let's start with calculating the similarity between doc1 and the query

	news	republican	candid	organ	campaign	presidenti	food	sim
doc1	.2	.3	.1					
query	1	1			1	1		

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n x_i^2} * \sqrt{\sum_1^n q_i^2}}$$

Similarity between doc2 and the query

	news	republican	candid	organ	campaign	presidenti	food	sim
doc2	.01	0	0	.4	.3	0	.3	
query	1	1	0	0	1	1	0	

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n x_i^2} * \sqrt{\sum_1^n q_i^2}}$$

Similarity between doc3 and the query

	news	republican	candid	organ	campaign	presidenti	food	sim
doc3	.15	.35			.4	.35		
query	1	1	0	0	1	1	0	

$$\text{sim}(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n (x_i * q_i)}{\sqrt{\sum_1^n x_i^2} * \sqrt{\sum_1^n q_i^2}}$$

HOW ABOUT THE OTHER DOCS?

$\text{sim}(\vec{d1}, \vec{q}) =$

$\text{sim}(\vec{d2}, \vec{q}) =$

$\text{sim}(\vec{d3}, \vec{q}) =$

$\text{sim}(\vec{d4}, \vec{q}) =$

$\text{sim}(\vec{d5}, \vec{q}) =$

SUMMARY OF SIMILARITIES:

	news	republican	candid	organ	campaign	presidenti	food	sim
doc1	.2	.3	.1					0.668
doc2	.01			.4	.3		.3	0.2658
doc3	.15	.35			.4	.35		0.9559
doc4	.25	.32	.15		.33	.4		0.9622
doc5	.1			.43	.3		.5	0.2735
query	1	1			1	1		

NOW COMPARING THE SIMILARITIES:

	sim
doc1	0.668
doc2	0.2658
doc3	.9559
doc4	.9622
doc5	.2735

Returned in this order:

doc4

doc3

doc1

doc5

doc2

VECTOR SPACE COMPARISON ADVANTAGES:

- Documents can be found which are most similar to the query without the need for a 100% match
- Returned documents can be sorted in decreasing order of similarity to query (so we have some ranking)
- Most commonly used approach across search engines and applied widely elsewhere also

QUESTION:

Are query vector weights always 1?

No Usually query terms are expanded and terms are weighted according to:

- Whether term is an original part of query or whether it was added by the search engine (e.g., as part of thesaurus look up for example or as part of personalisation information)
- Whether term has been used previously by that person (personalisation information) and whether the term is currently being used by other people (popularity + personalisation).

EXAMPLE 2

Given the vector tf*idf representation calculated of 3 documents with terms sql, database, program, comput

Find the most relevant document to the query “**database
programming with sql**” which is represented by the following query ‘database program sql’ and the vector query:

$\langle 1, 1, 1, 0 \rangle$

* Use precision to 4 decimal places for final answer

RECALL: tf*idf WEIGHTS (to precision of 3 decimal places)

	d1	d2	d3
sql	0.081	0.024	0.086
database	0.023	0.088	0
program	0	0.047	0.014
comput	0.021	0	0.019

Now adding in query vector (assume weights of all terms are 1, given that we are not told otherwise)

	d1	d2	d3	query
sql	0.081	0.024	0.086	1
database	0.023	0.088	0	1
program	0	0.047	0.014	1
comput	0.021	0	0.019	0

Can now start to calculate similarities:

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n x_i * q_i}{\sqrt{\sum_1^n x^2} * \sqrt{\sum_1^n q^2}}$$

NOTE:

The same approach can be used to determine how similar documents are to each other

Where might this be useful?

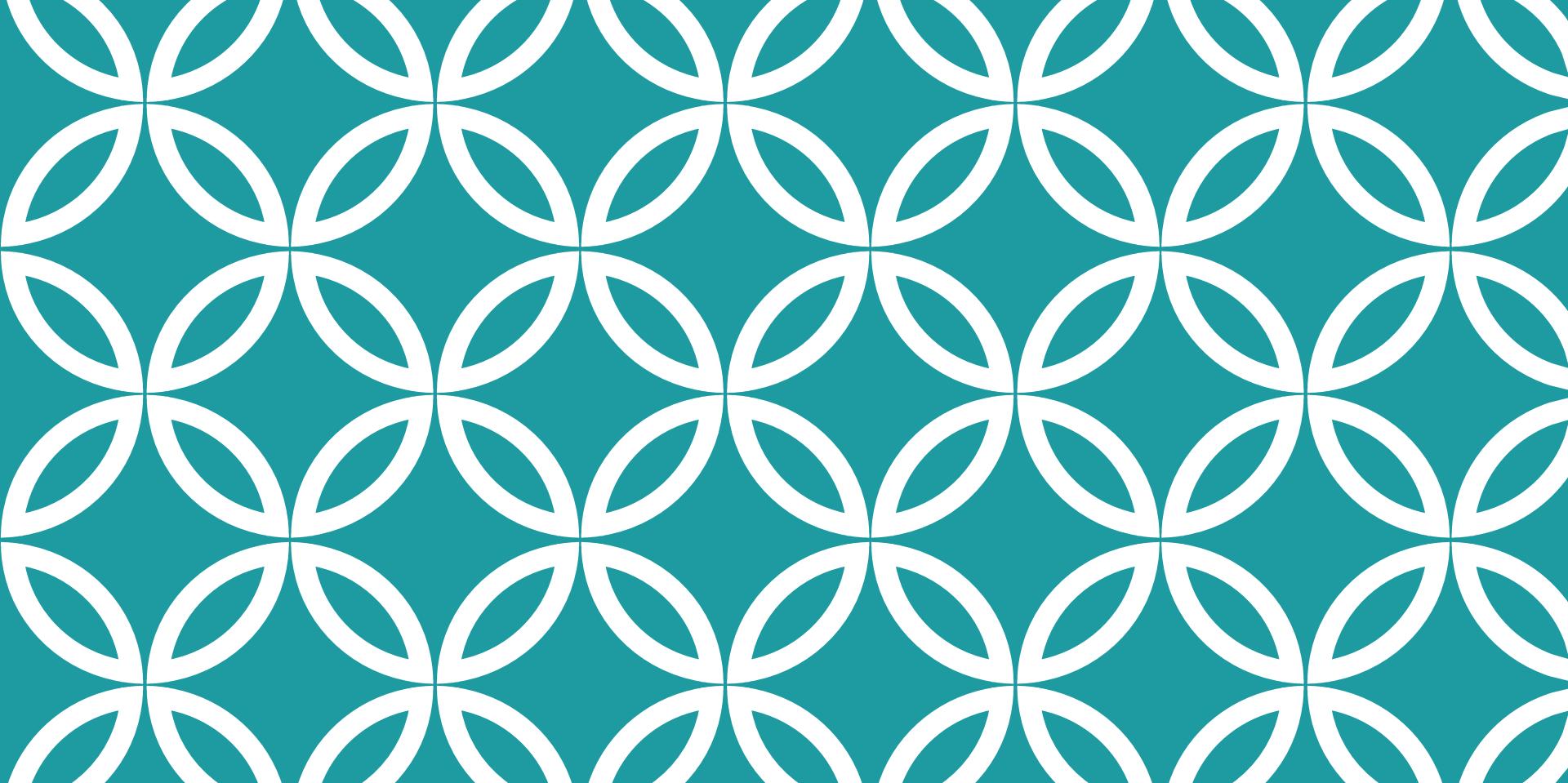
	d1	d3
sql	0.081	0.086
database	0.023	0
program	0	0.014
comput	0.021	0.019

For example, the similarity between d1 and d3?

SUMMARY:

$$sim(\vec{d}, \vec{q}) = \frac{\vec{d} * \vec{q}}{|\vec{d}| * |\vec{q}|} = \frac{\sum_1^n x_i * q_i}{\sqrt{\sum_1^n x^2} * \sqrt{\sum_1^n q^2}}$$

- Matching to find similar documents is usually performed using the dot product of the vector representations of documents and queries
- Entries with 0 can be ignored
- The vector norm (denominator) can be pre-calculated for all documents



PAGE RANK

(Finding Authoritative Web Pages)

CT102

Recall:

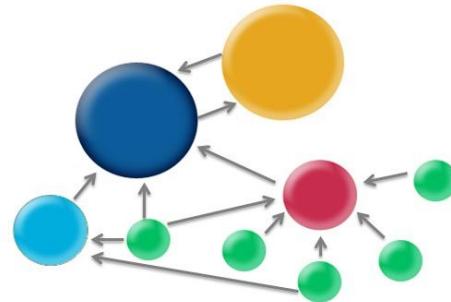
SERP (Search Engine Results PageS) and Web Page Ranking

Ranking involves ordering the results returned in response to a user query by one or more **scores** which are assigned to web pages by the search engine.

These scores can be assigned based on:

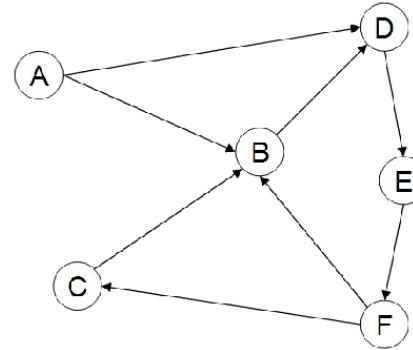
- Similarity scores (organic – index of crawled web pages and user query at search time)
- Potentially Page rank scores (organic – using links-pre-calculated)
- Ad word scores (sponsored - at search time)
- Personalised Information (Language, Location, Previous web searches, Browsing history, Tracking information, etc.)
- Featured articles/QA/Knowledge Graph information

PAGE RANK



- Page rank uses a **link analysis** algorithm to determine the **relative importance** of each web page.
- Page rank was originally developed by the founders of Google and was one factor which led to Google's dominance initially.
- The approach is currently used in Google and other search engines.
- Page rank only considers the in and out (hypertext) links of a web page (not the content, and not internal links).
- Each web page can be assigned a **PageRank score**.

OVERVIEW



- **Idea:** The higher the page rank score the more important the page and such pages should be ranked above pages with lower page rank scores (if both pages are returned in response to a query and are otherwise equally relevant to the query)
- The general idea is that a link from a web page is a “vote” or “endorsement” of the page it links to.
- The Page Rank algorithm tries to “sum up” all the votes for pages.
- However each page that “votes” also has its own Page Rank score and spreads its vote over all the pages it links to.
- Uses a **link analysis** algorithm to determine the **relative importance** of each web page

From the original Page Rank paper

<http://infolab.stanford.edu/~backrub/google.html>

“We assume page A has pages T₁...T_n which point to it. The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$\text{PR}(A) = (1-d)/N + d * (\text{PR}(T_1)/C(T_1) + \dots + \text{PR}(T_n)/C(T_n))$$

.... (For N web-pages,) PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web”

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page
(sergey, page)@cs.stanford.edu
Computer Science Department, Stanford University, Stanford, CA 94305

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure of the web hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype uses a flat text and hyperlink database of at least 24 million pages as a model of the World Wide Web.

To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advances in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper presents our approach to building such an engine.

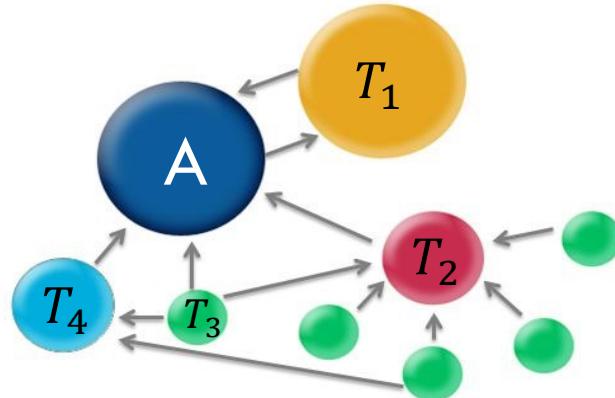
Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

Keywords: World Wide Web, Search Engines, Information Retrieval, PageRank, Google

1. Introduction

Note: There are two versions of this paper – a longer “for version control” longer “printable version”. The full version is available on the web and for conference CD-ROM. The web version has some challenges to information retrieval. The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web research. People are likely to surf the web using its link graph, often starting with high-quality human maintained indices such as Yahoo, or with search engines. Human maintained lists cover popular topics effectively but are subjective, expensive to build and maintain, slow to improve, and cannot cover all exotic topics. Automated search engines that rely on keyword matching usually return too many low-quality matches. To make matters worse, some advertisers attempt to gain people’s attention by taking measures meant to mislead automated search engines. We have built a large-scale search engine which addresses many of the problems of existing systems. It makes especially heavy use of the additional structure present in hypertext to produce much higher quality search results. We chose our system name, Google, because it is a common spelling of googol, or 10¹⁰⁰ and fits well with our goal of building very large-scale search engines.

More simply ...



If A is a web page linked to by 4 other web pages, T_1, T_2, T_3, T_4 and $C(T)$ is the total number of links **from** a web page T (the *outlinks* of page T) then the Page Rank of A, for N pages $PR(A)$ is =

$$(1-d)/N + d * (PR(T_1)/C(T_1) + PR(T_2)/C(T_2) + PR(T_3)/C(T_3) + PR(T_4)/C(T_4))$$

In general for any web page A with web pages $T_1 \dots T_R$ linking to it

... and also knowing for each web page, $C(T)$, the total number of links from a web page T (*outlinks*), and N the total number of web pages:

The Page Rank of A is defined as:

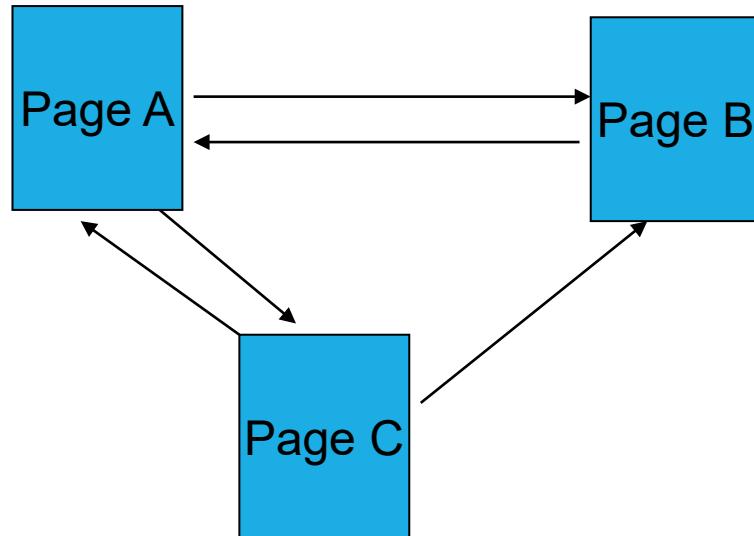
$$PR(A) =$$

$$(1-d)/N + d * (PR(T_1)/C(T_1) + \dots + PR(T_R)/C(T_R))$$

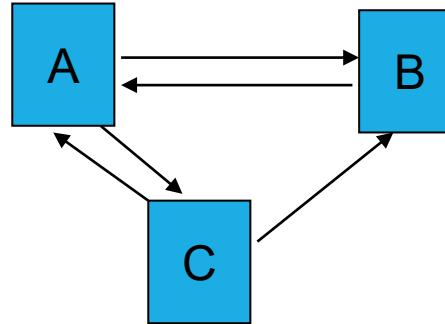
To get started

- We do not know the page rank of the pages to use – this needs to be calculated.
- Each page must be given an initial (estimate/guess) of a Page Rank score (usually 1 or $1/N$)
- This value is initially used as a guess for all PR values needed (all pages)
- This is modified over a number of iterations until it converges (settles) to some value, i.e. it only changes by a very small number, such as .0001, (if at all), from one iteration to the next
- For each iteration, the PR scores of each page from the previous iteration is used in the calculations
- d is usually set to 0.85

EXAMPLE 1 Calculate the page rank scores for each of the 3 web pages in the following graph. Take $d = 0.85$

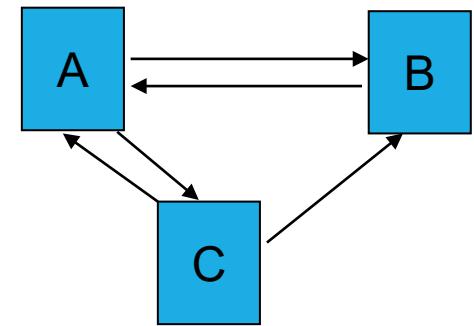


INITIAL STEPS:



- Count $C()$ for each page: $C(A) = 2$, $C(B) = 1$, $C(C) = 2$
- For web page A: vote comes from B and C
- For web page B: vote comes from A and C
- For web page C: vote comes from A only
- Assign initial PRs: $PR(A)=PR(B)=PR(C)=1/3$
- Set $d = 0.85$
- Write the formula for each page:
- $PR(\text{page}) = (1-d)/N + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn))$
- Keep calculating the formula for each page using the scores from previous iterations until convergence is reached

Write the formula for each page and start calculating:



- $\text{PR}(\text{page}) = (1-d)/N + d (\text{PR}(T1)/C(T1) + \dots + \text{PR}(Tn)/C(Tn))$
- Store the PR result from the previous calculation in `prevA`, `prevB`, `prevC`
- $\text{prA} = 0.15/3 + 0.85 * (\text{prevA}/cB + \text{prevC}/cC);$
- $\text{prB} = 0.15/3 + 0.85 * (\text{prevA}/cA + \text{prevC}/cC);$
- $\text{prC} = 0.15/3 + 0.85 * (\text{prevA}/cA);$
- $\text{prevA} = \text{prA};$
- $\text{prevB} = \text{prB};$
- $\text{prevC} = \text{prC};$

WRITING IN C CODE

```
prA = prB = prC = 1.0/3.0;  
prevA = prevB = prevC = 0.0;  
cA = 2;  
cB = 1;  
cC = 2;  
for(i=0; i < 10; ++i)  
{  
    prevA = prA;  
    prevB = prB;  
    prevC = prC;  
    prA = 0.15/3 + 0.85 * (prevB/cB + prevC/cC);  
    prB = 0.15/3 + 0.85 * (prevA/cA + prevC/cC);  
    prC = 0.15/3 + 0.85 * (prevA/cA);  
}
```

CALCULATE IN EXCEL OR EQUIVALENT:

C(A)	C(B)	C(C)
2	1	2
PR(A)	PR(B)	PR(C)
$0.15/3 + 0.85*(PR(B)/C(B) + PR(C)/C(C))$	$0.15/3 + 0.85*(PR(A)/C(A) + PR(C)/C(C))$	$0.15/3 + 0.85*(PR(A)/C(A))$
0.333333333	0.333333	0.333333333
0.475	0.333333	0.191666667
0.414791667	0.333333	0.251875
0.440380208	0.333333	0.226286458
0.429505078	0.333333	0.237161589
0.434127008	0.333333	0.232539658
0.432162688	0.333333	0.234503979
0.432997524	0.333333	0.233669142
0.432642719	0.333333	0.234023948
0.432793511	0.333333	0.233873156
0.432729424	0.333333	0.233937242

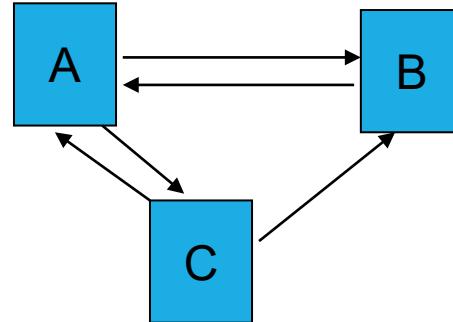
RESULTS (to 4 decimal places):

After 10
iterations:

$$\text{PR(A)}=0.4327$$

$$\text{PR(B)}=0.3333$$

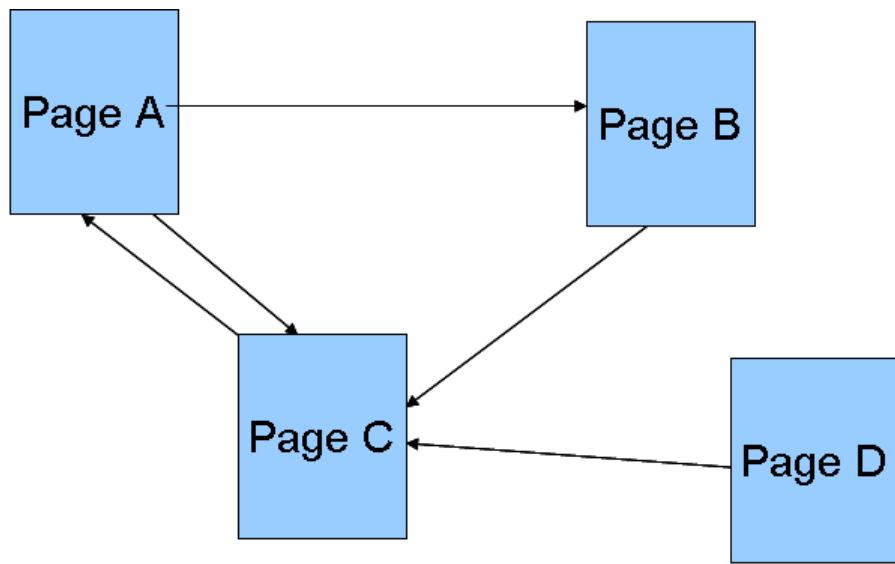
$$\text{PR(C)}=0.2339$$



How many iterations are required?

- Usually compare the difference between the last two values of the PR scores per web page (e.g. prA and prevA)
- When this difference is small, e.g., .0001 can stop
- In general the number of iterations will depend on the number of web pages
- For a small number of web pages can set the number of iterations (e.g., 10 in previous example)
 - For web indexed pages (millions of web pages), can converge in 50 to 60 iterations

YOU TRY: EXAMPLE 2: Calculate the page rank scores for each of the 4 web pages in the following graph. Take $d = 0.85$



Steps:

1. Specify the inlinks and the number of outlinks for each webpage
2. Write the formula for each web page
3. Code and get results to 4 decimal places

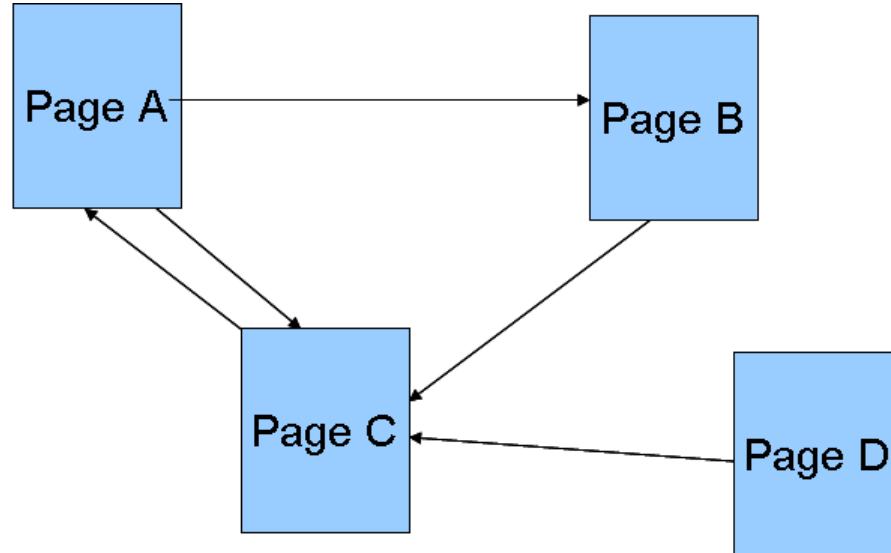
Answer after 14 iterations (to 4 decimal places)

$PR(A) = 0.3722$

$PR(B) = 0.1959$

$PR(C) = 0.3944$

$PR(D) = 0.0375$



NOTES:

- According to the Page Rank formula, page D gets a Page Rank value (of 0.0375) even though it has no links to it
- In reality pages with no links will be discarded from the calculation at the start and just given the $1-d/N$ score at the start

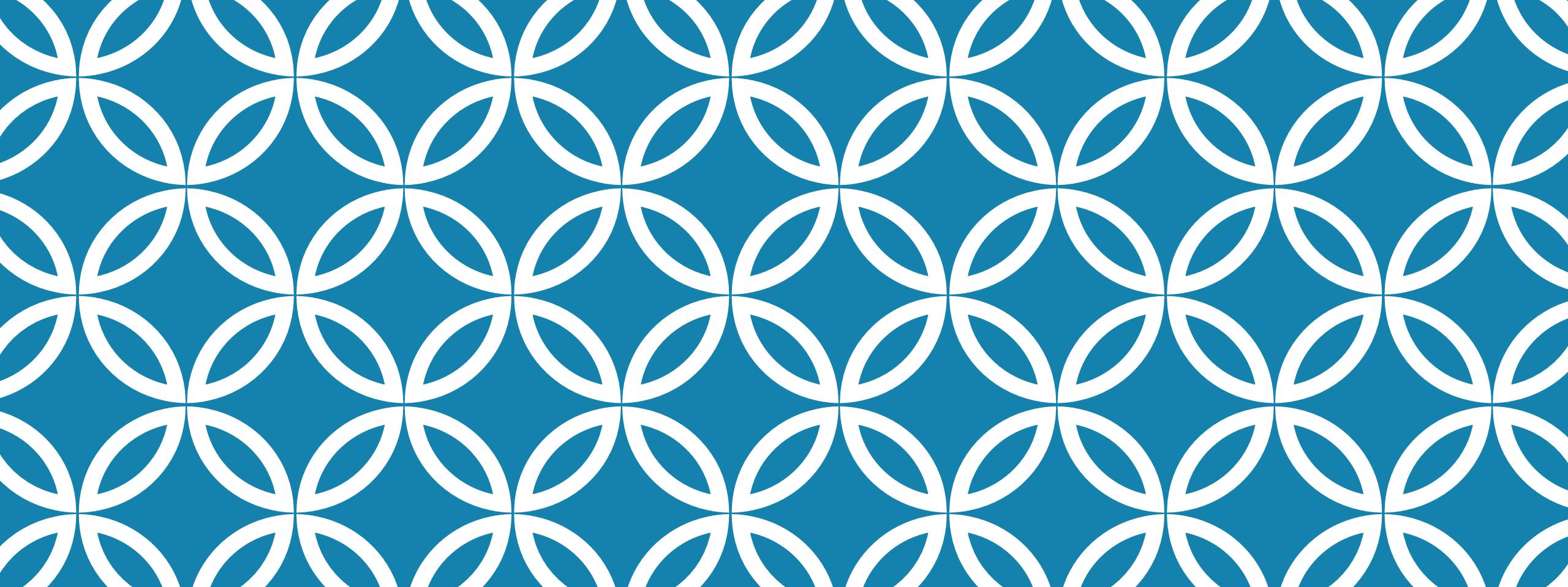
SUMMARY

The page rank approach is a common algorithm used by search engines to find *authoritative* or more important (organic) web pages. It uses the web link structure – and no content.

It can be seen as a measure of *crowdsourced quality* of a page relative to other pages.

It is independent of any query and any personalisation.

It can be pre-calculated based on the information gathered by the crawler and stored in the index (i.e., a PR score is associated with each page in the index).



WEB SEARCH:

WEB SEARCH: ADS AND PERSONALISATION

CT102
Information
Systems

SERP (Search Engine Results PageS) and Web Page Ranking

Ranking involves ordering the results returned in response to a user query by one or more **scores** which are assigned to web pages by the search engine

These scores can be assigned based on:

- Similarity scores (organic – index of crawled web pages and user query at search time)
- Potentially Page rank scores (Google) (organic – using links-pre-calculated)
- Ad word scores (sponsored - at search time)
- Personalised Information (Language, Location, Previous web searches, Browsing history, Tracking information, etc.)
- Featured articles/QA/Knowledge Graph information

ADS



Most web search engines have a model of “[paid-for-ads](#)” where web pages from this category are ranked above “[organic results](#)” (results ranked according to the other scores) once there is a strong similarity (match) between the terms the ad has picked as “ad words” and the query terms.

Thus an [additional search](#) occurs independent of the search using the indexed representation of web documents.

This additional search uses the [repository \(database\)](#) of ad words and a [scoring](#) of the ads such that ads are ranked and placed before the organic ranked results.

Examples: Compare searching for “nike runners” and “running shoes review”

Google X Microphone Search

All Images Shopping News Videos More Tools

About 209,000,000 results (0.76 seconds)

Ad · <https://www.nike.com/running/shoes> Nike Running Shoes - Nike Running Collection Speed And Stamina. Get Back What You Put In With The Latest Nike Running Tech. Get The Support And Comfort You Need To Feel Your Best Every Stride With Nike.

Women's Running
Run In Style With Womens
Running Products At Nike.com

Men's Running
Unparalleled Running Technology.
Explore Running For Men At Nike.com

Ad · <https://www.lifestylesports.com/nike/runners> Nike Runners - Free Next Day Delivery Enjoy Free Next Day Delivery On All Orders Over €50 And Free Fast Returns. Everything You Need For Your Life. Your Style. Your Sports.

Ad · <http://www.elverys.ie/training/2021> Nike Runners - Free Delivery Orders Over €50 - elverys.ie Explore our running & training range from leading brands, delivered across Ireland. Get free delivery on orders over €50 across Ireland. Shop online or in-store today!

Ad · <https://ie.sportsdirect.com/> Nike Running Shoes - Sale - Up To 70% Off Up To 70% Off Online Sale - Use Click & Collect And Get A €10 Voucher With €100 Spend!

<https://www.nike.com/running-shoes-37v7jzy7ok> Running Shoes & Trainers. Nike IE Choose from dozens of Nike running shoes to find the perfect pair for your style and skill level.

<https://www.nike.com/running> Nike Running. Nike IE Recover better and come back stronger. Shop Running Shoes By Surface. Running Shoes - Women's Running Shoes - Nike Zoom Fly 3 - Sale Running

<https://www.nike.com/> Nike Sale. Get Up To 50% Off. Nike IE Men's Easy On/Off Road Running Shoes (Extra Wide). 1 Colour. €83.97. €119.99 ... Nike Air Zoom Pegasus 38 Older Kids' Road Running Shoes. Sale Running - Shoes - Men's Sale - Sale Hoodies & Sweatshirts

Google X Microphone Search

All Videos Images Shopping News More Tools

About 196,000,000 results (0.72 seconds)

Ads · Shop now

	NIKE REVOLUTION 5... €55.00 Elverys.ie ★★★★★ (5k+) By Feedopti...		Men's Nike Revolution 5... €60.00 Life Style Sports Free delivery By Google
	Nike React Infinity Run... €79.97 €60 Nike Official ★★★★★ (822) By Pricesear...		Nike Revolution 5 Men's... €48.00 Sports Direct Ire ★★★★★ (822) By Buy Bye
	NIKE REVOLUTION 5... €35.00 Elverys.ie ★★★★★ (2k+) By Pricesear...		Women's Nike Revolution 5... €50.00 €60

Ads · Shop running shoes review

	ASICS NOVABLAST... €70.00 €140 Elverys.ie Free delivery By Feedopti...		Men's Nike Zoomx Invincible... €110.00 €180 Life Style Sports Free delivery By Google
	Saucony Endorphin Shift... €106.46 SportsShoes.con ★★★★★ (119) By Redbrain		New Balance Fresh Foam... €99.99 SportsShoes.con ★★★★★ (870) By Redbrain
	Asics Gel Pulse 11 Running... €79.55 SportsShoes.con ★★★★★ (1k+) By Redbrain		

<https://www.runnersworld.com/gear/best-running-sho...> Best Running Shoes | Running Shoe Reviews 2021 22 Jul 2021 — SOFT - Hoka One One Mach 4 - Nike ZoomX Invincible Run Flyknit - New Balance Fresh Foam 1080 v11 - Brooks Glycerin 19 - Asics EvoRide 2 - Adidas ...

<https://www.runningshoesguru.com/reviews> 933 Running Shoes Reviews (September 2021) The best Running Shoes Reviews on the internet! Our testers run and analyze all the latest shoes - and you can read hundreds of feedback from our readers! All Nike - ON Running (11) - Running Shoes Reviews - Brooks Aurora Review

<https://runrepeat.com> RunRepeat: Reviews of Running Shoes, Hiking, Training ... Shoe reviews from 1 million users & 1000 experts. Running, sneakers, training, hiking, soccer, basketball. Trusted & independent. Best price guarantee! 10 Best Running Shoes in 2021 - Trail running shoes - Golf shoes

<https://runrepeat.com/catalog/running-shoes> 1000+ Running shoes - Save 36% | RunRepeat Running shoes - On Cloud. 89. Great (24,679 reviews) - Adidas Ultraboost. 89. Great (36,865 reviews) - Brooks Adrenaline GTS 21. 93. Superb (44,367 reviews).

People also ask

Which brand is best for running shoes?

Are on running shoes actually good?

HOW DOES IT WORK?

Each ad is associated with **keywords** (submitted by the advertiser).

Advertisers also indicate how much they are willing to pay for an ad to be displayed – this is the advertiser's **bid**.

The Ads system checks if an ad's keywords match the terms in a query.

If so, then the ad is considered eligible to appear in the search results.

The ad then goes through an "**Ad Auction**".

The auction involves bidding and determines whether or not the ad is actually displayed.

AD AUCTION

How do ads win based on their bids?



Based on 2 (or more) components:

1. **Max Bid value** – max amount advertiser will pay the search engine for click (CPC – cost per click) on the ad. Advertiser will pay a CPC if the ad is displayed and clicked.
2. **Quality of ad** – relevance and usefulness of ad and web page it links to. Quality is calculated based on 3 aspects.

Also (in Google's case at least):

3. **“Expected impact of ad extensions”** where ad extensions are extra information provided (in addition to the keywords).

QUALITY OF ADS

2.1: expected CTR (Click Through Rate): prediction on how often an ad will be clicked on when displayed – based on how ad preformed previously when displayed for same/similar query.

2.2: Ad relevance: keywords of ad versus keywords used in query.

2.3: Landing page experience: quality and relevance of page ad links to – should be related to query, should be “good quality” website/webpage.

Some researchers have found that the weighting used by Google is:

- Expected CTR: 39%
- Ad Relevance: 22%
- Landing page experience: 39%

EXAMPLE (from Google)

4 ads (A, B, C, D)

3 components:



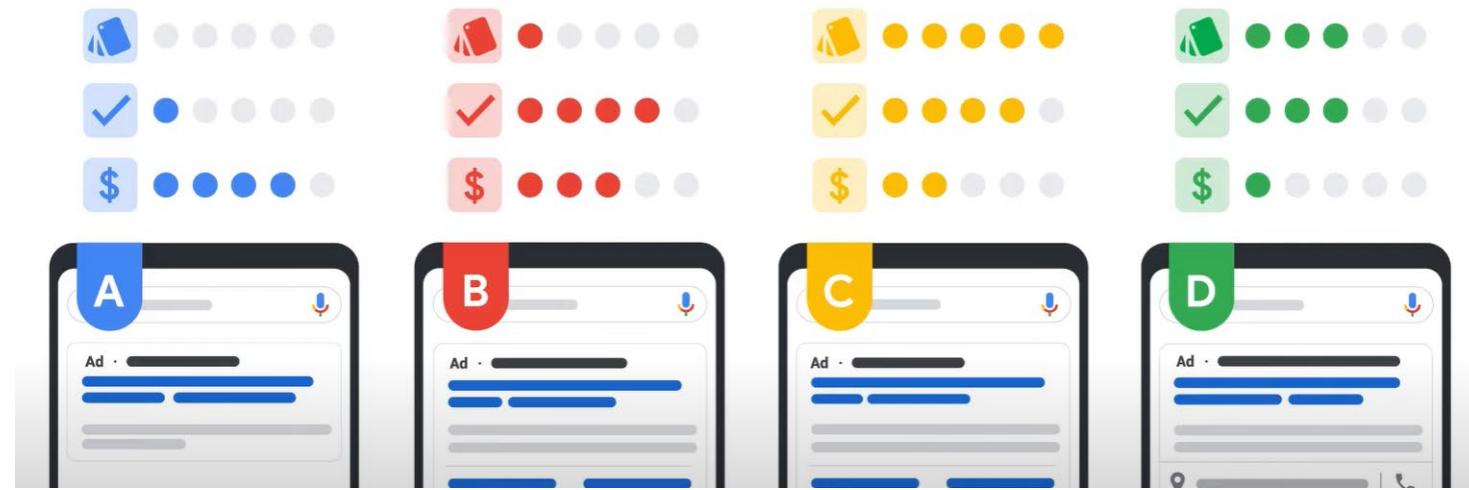
Impact



Quality



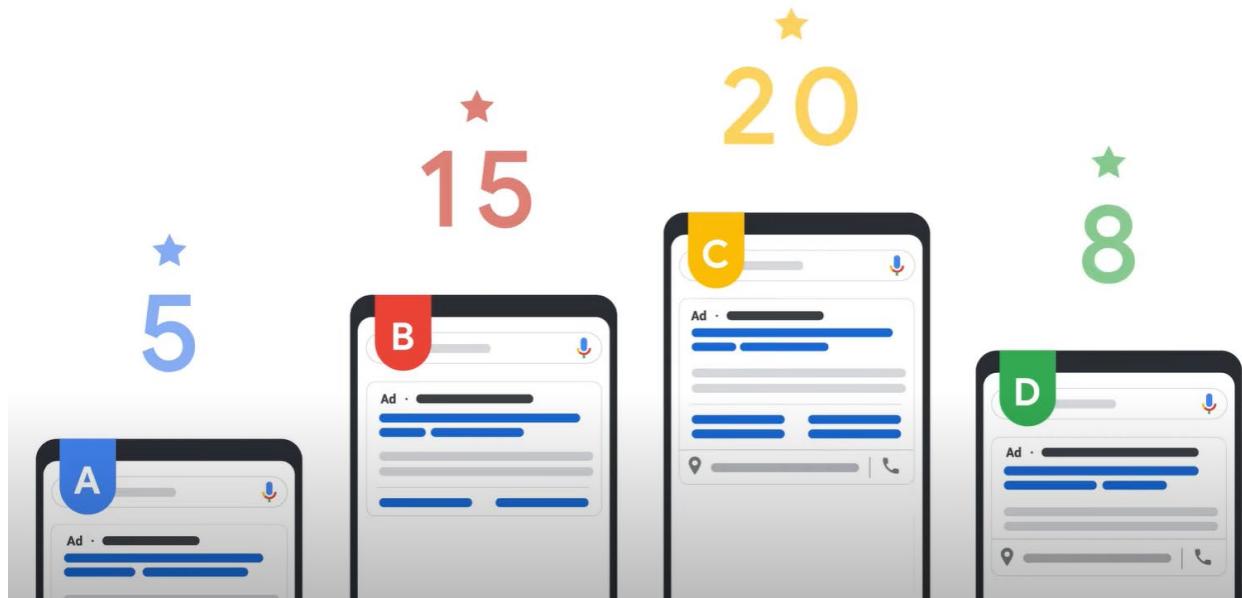
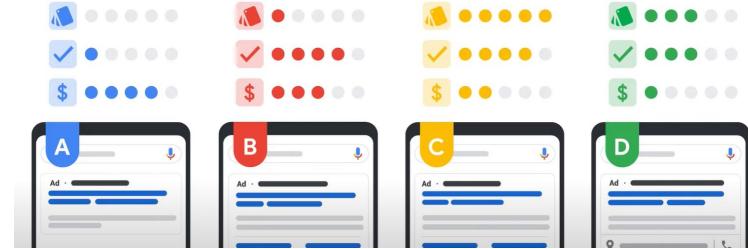
Bid Value



Where quality =
expected CTR
ad relevance
landing page experience

EXAMPLE (from Google)

4 ads (A, B, C, D)



Ad rank values: 5, 15, 20, 8

So ranking of ads displayed = C, B, D

(and probably not A as rank value too low and/or only displaying top 3 ads)

CPC: COST PER CLICK

HOW MUCH DOES AN ADVERTISER ACTUALLY PAY?

Based on the ads ranking, its quality score (which incorporates the max bid) and some standard cost:

For example, for some ad which will be displayed in the first position

- $\text{CPC}(\text{ad}) = \text{rankValue}(\text{ad in second position})/\text{quality score}(\text{ad}) + \text{cost}$

Ad ID	Max Bid €	Quality Score	AdRank Value	Rank	CPC Calculation	Actual CPC €
adR	4	8	32	1	$27/8+0.01$	3.39
adX	3	9	27	2	$24/9+0.01$	2.68
adY	6	4	24	3	$16/4+0.01$	4.01
adP	8	2	16	4		

“Bad Ads”

The screenshot shows a news article from Mint. At the top, there's a navigation bar with links like 'ACM Summer School', 'NUIG Research', 'project', 'Text By the Bay 201...', and 'Learn Italian:'. Below the navigation is the main content area with a large heading 'Google says it removed over 3 billion bad advertisements globally in 2020'.

mint

Home > Technology > News > Google says it removed over 3 billion bad advertisements gl...

Google says it removed over 3 billion bad advertisements globally in 2020



FILE PHOTO: A man walks past a logo of Google (REUTERS)

1 min read . Updated: 19 Mar 2021, 08:10 PM IST

ANI

Tech giant Google recently revealed in a blog post that it blocked or removed 3.1 billion bad ads, including COVID-19 related advertisements, internationally in 2020 for violating its policies

SEM

Google removed 2.7 billion bad ads, nearly 1 million ad accounts in 2019

This year, the company says it has removed "tens of millions" of COVID-19 related ads.

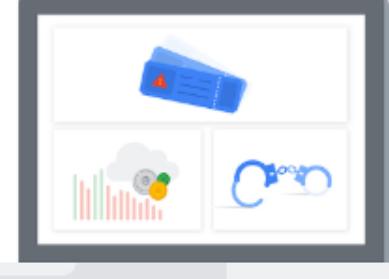
Ginny Marvin on April 30, 2020 at 1:00 am

Last year, Google says it took down 2.7 billion so-called bad ads for violating the company's ad policies, according to its [annual report](#) released Thursday.

That's up from the 2.3 billion bad ads Google reported taking down [in 2018](#). The number of ad accounts Google terminated remained relatively flat from the previous year at nearly one million.

Publisher network. Google also noted that it terminated the accounts of more than 1.2 million publishers and removed ads from over 21 million [web names](#) across its publisher network for policy

2.3B
bad ads taken down



Dozens of new ads policies to take down billions of bad ads

Bad ads, fake ads ...



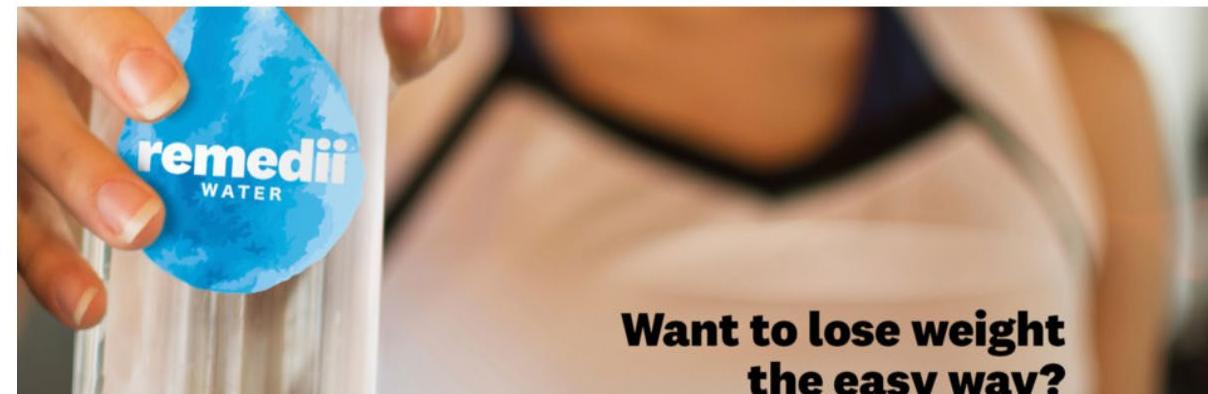
News

All ne

Scams & Fraud

Fake ads; real problems: how easy is it to post scam adverts on Facebook and Google?

We were able to promote fake health advice and a brand that didn't exist to highly targeted audiences online



How is this linked to Programmatic Advertising?

- Programmatic advertising automatically places “**contextually relevant advertisements**” on a website (not on web search page necessarily).
- Website webmaster includes (Javascript) code on webpage to indicate where ads can be placed.
- Each time site is visited, the code fetches ads from the relevant servers and displays these on the website.
- The ads fetched are based on a number of approaches but again have the same “auction” approach.
- Site as well as ad engine (e.g. Google) gets paid for the ads clicked/viewed – only works if there isn’t an ad blocker.
- Often website ad strategy from webmaster point of view is performed in conjunction with SEO (Search Engine Optimisation).

CLASS QUESTION?

Do you view personalised ads as generally useful or not?

Aside: Search Engine Optimisation (SEO)

Web search engines allow webmasters to provide site maps and instructions to crawler including request to recrawl or not

Webmasters try to ensure that their page will be indexed “well” by search engines and thus appear in a top set of returned results if relevant to a query:

Basic techniques used:

- Good content which is updated on a regular basis
- Using tags and meaningful concise names in tags (such as titles etc.)
- Using site maps and XML sitemap
- Moderate and remove spam from any comments section
- Don't overuse ads
- Analyse who is visiting site and time spent etc.



Many Issues with Programmatic Advertising

- Vetting the ad content coming in – basic checks automatically done and after the fact with “crowdsourcing”
- Placement of ads ... especially if placed on sites with controversial topics
- Fake news ads and “Bad ads”
- Payment models ... “marking own homework”
- Efficacy ... “billboards underwater”
- Personalisation monetising personal information gathered as part of search

Google

Rob Davies

Mon 20 Mar 2017 12.23 GMT
@ByRobDavies

Head of Google Europe apologises over ads on extremist content

Matt Brittin says company ‘needs to do more’ but declines to say whether it will actively seek out inappropriate material



Google

Samuel Gibbs

Thu 7 Dec 2017 09.47 GMT

Unilever

EU warns tech firms: remove extremist content faster or be regulated

European commission tells Facebook, Google, YouTube, Twitter and others that legislation is being considered if self-regulation continues to fail

Marmite maker Unilever threatens to pull ads from Facebook and Google

One of world's biggest advertisers says it will avoid platforms that 'create division'



Julia Kollewe
Mon 12 Feb 2018 17.36 GMT
[f](#) [t](#) [e](#) [...](#)
2,419 | 1,029

Web Search Personalisation

Web search personalisation can be defined as any action that uses **user's interests** and **preferences** to tailor or re-rank the results returned by a search engine.

It potentially uses many different approaches: information retrieval, data mining, social networks, recommender systems.

"It will be very hard for people to watch or consume something that has not in some sense been tailored for them."

Eric Schmidt, Google

AIM OF PERSONALISATION TECHNIQUES

Aim is to:

1. obtain **user context** and to use this user context to improve results returned (e.g., language, location, preferences, etc.)
2. predict what a user's information need is without the user having to explicitly state it (e.g., query auto-complete, predictions, "featured article" section)
3. decrease search ambiguity and return results *more likely* to be interesting to a particular user providing more effective and efficient information access

WHAT IS CONTEXT?

Context: Any information that can be used to characterize the situation or intent of an entity

Three main aspects of context with respect to web search:

1. User's short-term information need (query)
2. Semantic knowledge about the domain being investigated (using knowledge bases)
3. User's long-term interests (user profile)

PERSONALISATION DATA

- Language
- Location
- Web history:

Search history and settings

Browsing history and click-through data

Social data (from logging in to social media platforms etc)

SEARCH HISTORY

Refers to the list of queries the user has recently entered

Typically, the query and date along with IP address and cookie information, if available, is stored

Also can be used to infer/predict a user's interests and preferences

BROWSING HISTORY

Storing what is clicked in the search results returned (url, title, date) and using this as a measure of user interest or preference

Can be categorised in to **long and short term interests** and often more recent history is considered more important (an indicator of current interest)

Can also be categorised in to different user profiles relating to different topics

This information can be used to re-rank results based on particular sites if previous history shows that you tend to **prefer (click frequently)** on that site

Often data mining techniques are used to discover “patterns”

IMPLICATIONS 1: PRIVACY

Many issues yet to be resolved:

- Users need to feel in control of the personal information stored on them and on how it is used, e.g. on/off option
- Users need to be convinced it is worth giving something to get something e.g.,
 - personal information for better results?
 - personal information for better ads?
 - personal information allowing “tracking” across platforms?
- Is web search engine’s focus on personalisation for ad revenue rather than for quality search results?
- What do we give up in order to access free services?



IMPLICATIONS 2: ETHICS

- From a **law-enforcement** perspective there is a subset of people whose online activities are of interest re. illegal activities – terrorists, arms, drugs, human trafficking, gambling, etc. – how to gather data only on these and not the general public?
- From a **human-rights** perspective there is a subset of people whose online activities are also of interest to relevant non-democratic authorities but may result in imprisonment etc. if detected
- In Ireland and elsewhere we are seeing cases in the courts where search engine history and server logs etc. are being used as evidence in serious crimes
- GDPR has helped in stopping data harvesting and selling personal data in the EU

IMPLICATIONS 3: RIGHT-TO-BE-FORGOTTEN

In May 2014, the European Court Of Justice ruled that EU citizens have a “Right To Be Forgotten,” that they could request that search engines remove links to pages deemed private, even if the pages themselves remain on the internet

Report: 2 years in, 75 percent of Right to Be Forgotten asks denied by Google

More than 50 percent of requests come from Germany and the UK.

Greg Sterling on May 12, 2016 at 5:28 pm

Right to be forgotten

Sarah Marsh

@sloumarsh

Tue 24 Sep 2019 16.04 BST



307

'Right to be forgotten' on Google only applies in EU, court rules

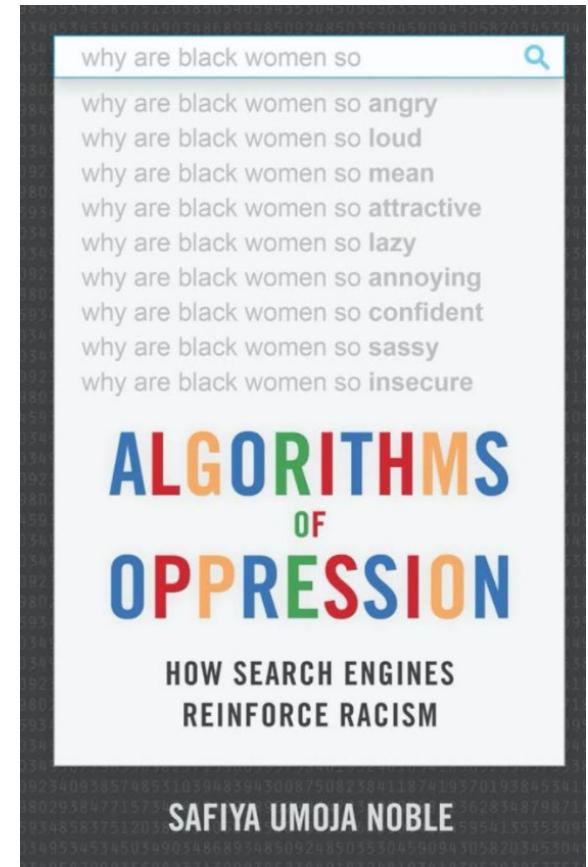
Europe's top court says firm does not have to take sensitive information off global search



▲ In 2015 France's privacy watchdog told Google to delist sensitive information from internet search results globally upon request. Photograph: Johannes Eisele/AFP/Getty Images

IMPLICATIONS 4: “FILTER BUBBLE” (Eli Pariser)

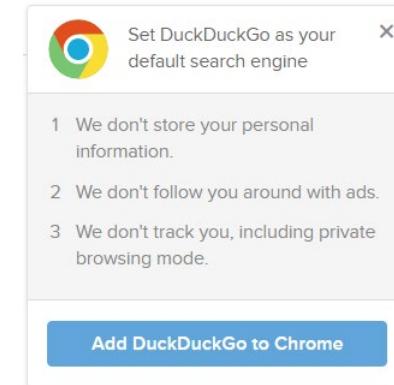
- Results returned to us are not unbiased, i.e., due to personalisation the same query from different people would give us different results.
- Different “views” of reality being filtered out – e.g., only seeing news similar to what you have seen in the past. Sometimes this may be fake information and filter bubbles mean you keep seeing more fake information.
- Shaping what people read = shaping what people think



SEARCH ENGINES THAT DO NOT TRACK

A number of search engines allow user's more control over the personal information that is used or do not use any personal information such as:

- Ask.com
- Startpage.com
- Duckduckgo.com

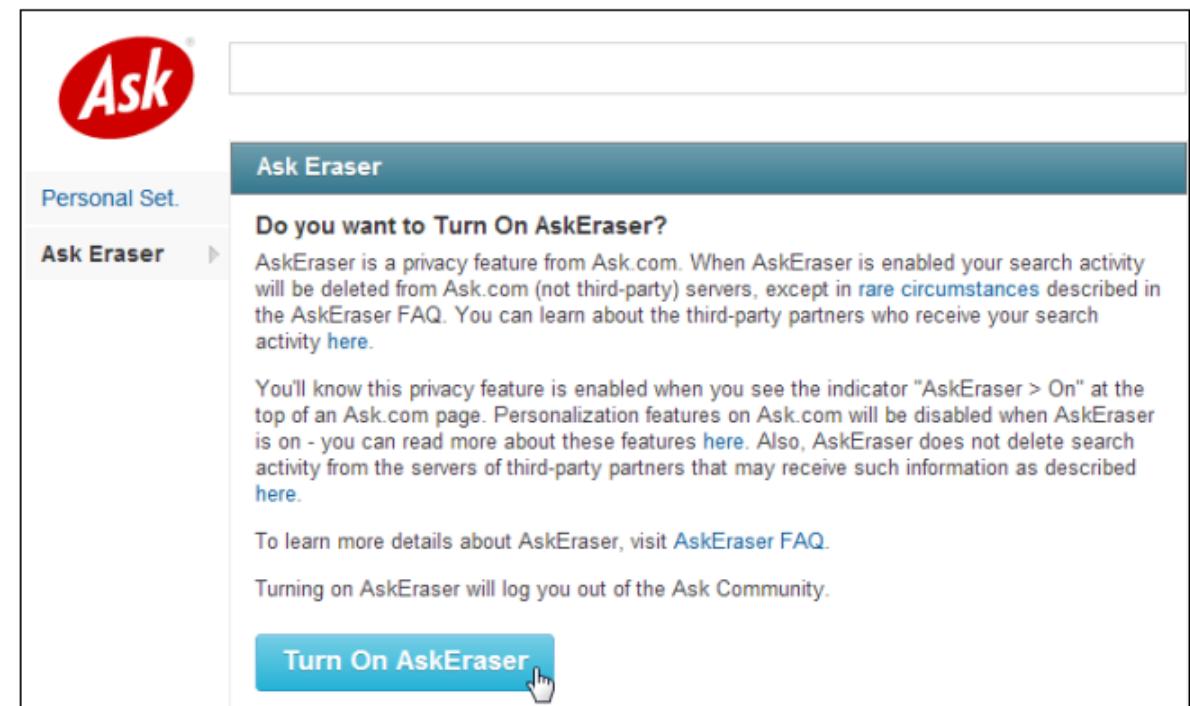


An interesting experiment is to pick a few different queries and compare results across a number of different search engines using the same queries.

ASK.COM

ask.com

Has an optional “Ask Eraser” which, when turned on, will delete old ask cookies, won’t store any new cookies and won’t keep search history “except in rare circumstances”



startpage

startpage.com

- Discards all personally identifiable information
- Discards IP addresses once search request complete
- Doesn't use cookies
- Doesn't keep a record of search queries
- Searches Google by submitting the user query to Google and displaying the results to the user
- All Google sees is a large amount of searches coming from Startpage's servers



DuckDuckGo



[Duckduckgo.com](https://www.duckduckgo.com)

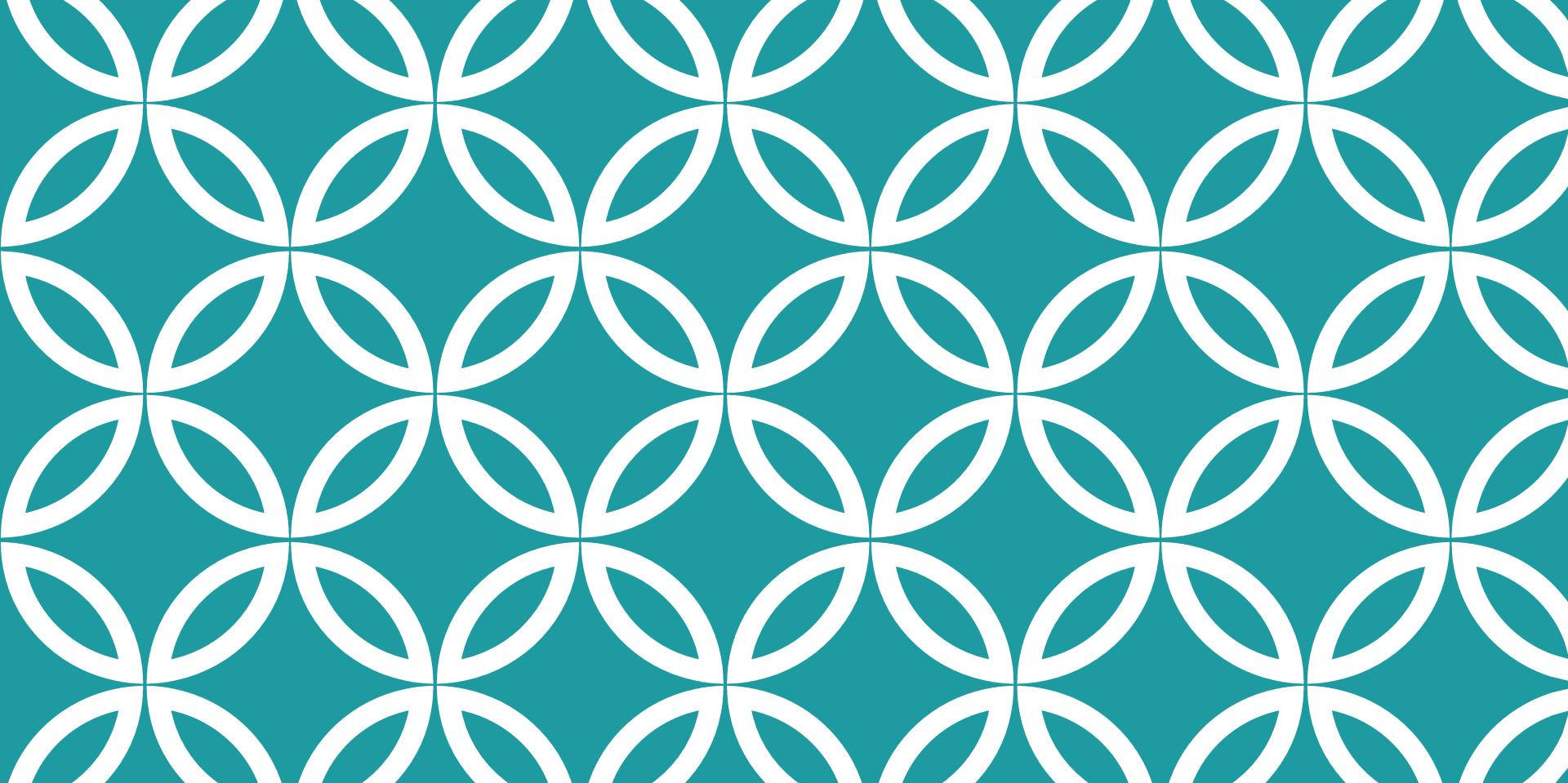
- Doesn't log any personally identifiable information
- Discards IP addresses etc. from its server logs
- Doesn't use cookies
- Doesn't generate an anonymized identifier to tie searches together
- As a result you will get the same results as everyone else using the same queries

SUMMARY

There are many algorithms at work behind the web search scene – some with respect to sponsored content (ads), some with respect to personalisation and tracking – and it is not always clear or obvious what is collected and stored and what/how it is used – and who it is benefiting.

Important to know about:

1. how the ad auction algorithms work.
2. the web search goal of gathering personal information – and the type of personal information that is gathered – and how this can be exploited elsewhere



DATABASE SYSTEMS

CT102:
Information
Systems

DATABASE SYSTEMS

employee											
	FName	MII	LName	SSN	BDATE	Address	SI	SALARY	SUPERSSI	DNO	
John	B	Smith	123456789	09/01/1965	731 Fondren, Houston, TX	M	€30,000.00	333445555	5		
Franklin	T	Wong	333445555	08/12/1955	638 Voss, Houston, TX	M	€40,000.00	888665555	5		
Joyce	A	English	453453453	31/07/1972	5631 Rice, Houston, TX	F	€25,000.00	333445555	5		
Ramesh	K	Narayan	666884444	15/09/1962	975 Fire Oak, Humble, TX	M	€38,000.00	333445555	5		
James	E	Borg	888665555	10/11/1937	450 Stone, Houston, TX	M	€55,000.00		1		
Jennifer	S	Wallace	987654321	20/06/1941	291 Berry, Bellaire, TX	F	€43,000.00	888665555	4		
Ahmad	V	Jabbar	987987987	29/03/1969	980 Dallas, Houston, TX	M	€25,000.00	987654321	4		
Alicia	J	Zelaya	999887777	19/07/1966	3321 Castle, Spring, TX	F	€25,000.00	987654321	4		

A database system is an Information System that stores and retrieves structured data

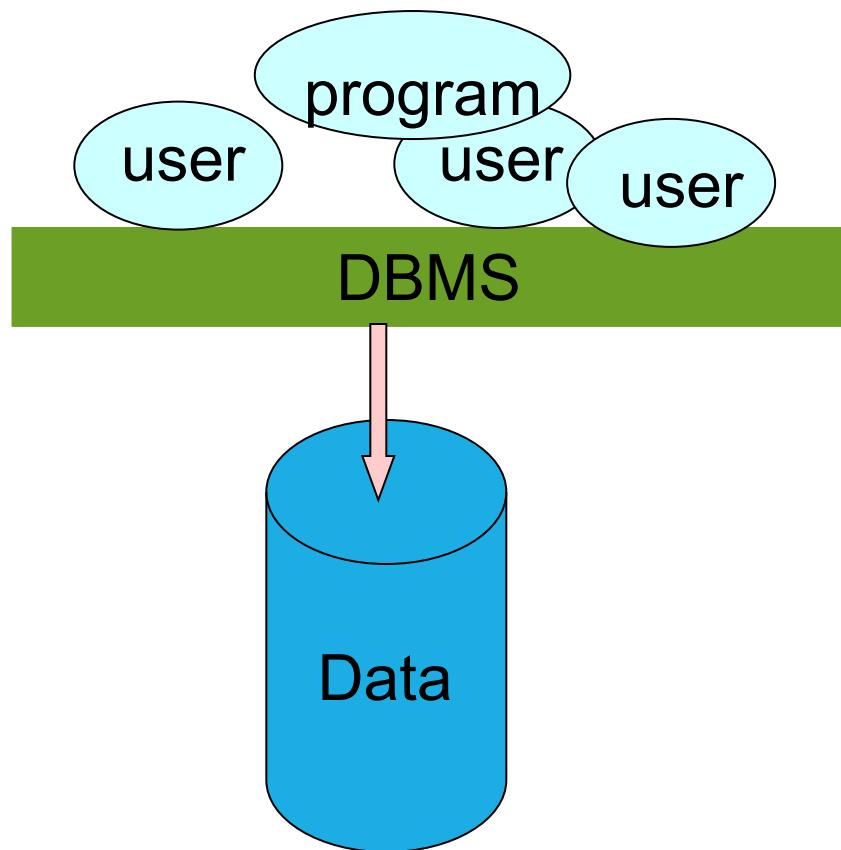
DATABASE DEFINITION

One or more **tables**

- where a table is an ordered collection of **records**
- where a record consists of data

DATABASE APPROACH

A single repository of data is maintained that is defined once and then accessed by various users/programs through a DBMS

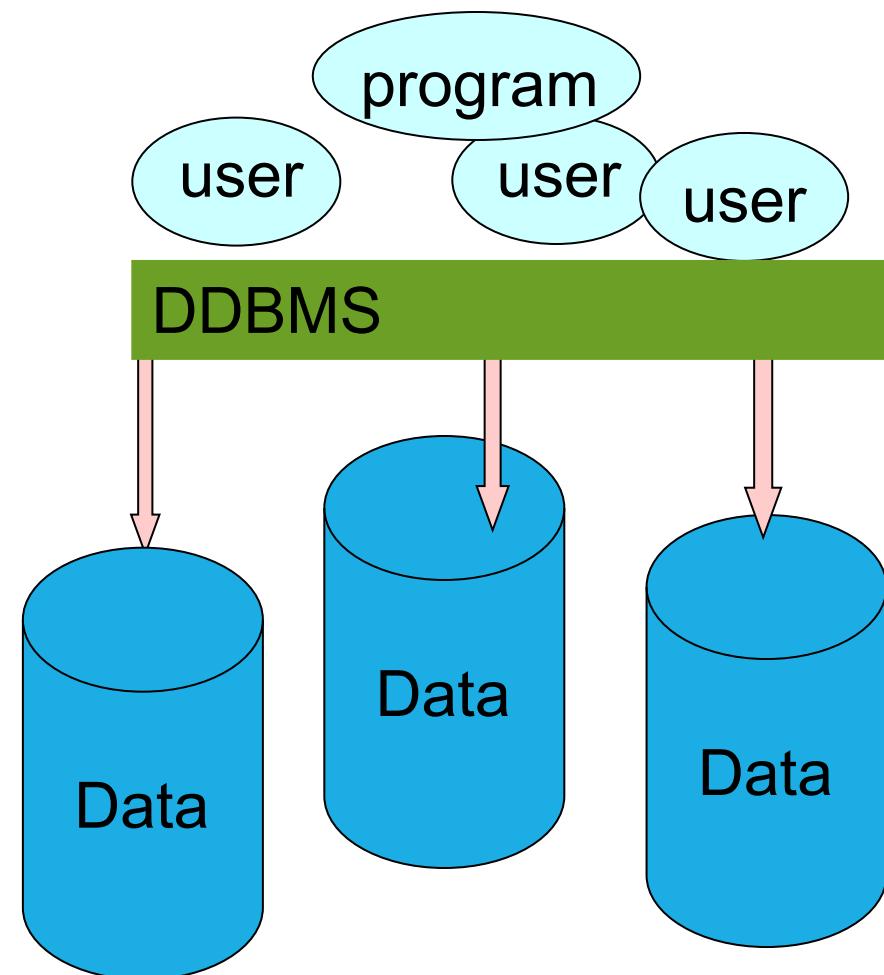


DISTRIBUTED DATABASE APPROACH

Data is defined once and then stored at multiple (**distributed**) sites

However:

Users have the impression of a single repository of data



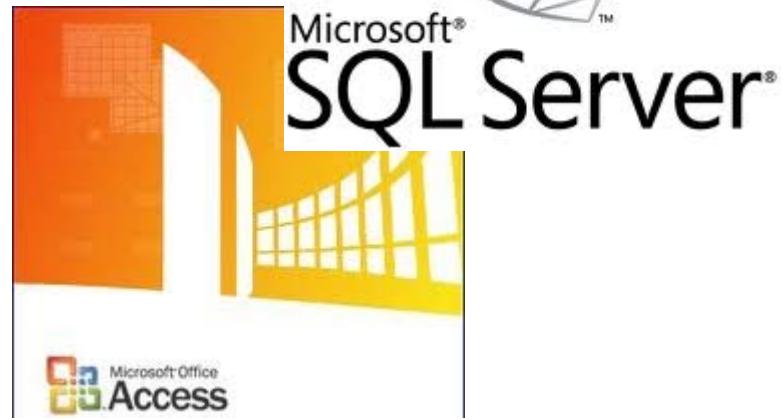
TYPES OF DATABASE SYSTEMS

- Relational Databases (mySQL, Sybase, Oracle)
- Non-Relational Databases (MongoDB, Redis, Apache Cassandra)
- XML databases (BaseX, eXist, Sedna)
- Blockchain databases



RELATIONAL DATABASE MANAGEMENT SYSTEMS (DBMS)

A DBMS is a collection of programs that facilitates the process of **defining, constructing** and **manipulating** databases for various applications.



ORACLE®



RELATIONAL DATABASES

Column Index							
Row Index	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							
7							

Based on the mathematical theory of relations
(Codd, IBM, 1970s)

Can be seen as consisting of “tables and only tables”:

- A table is a natural representation of a relation
- A table is a 2-d array

RELATION TABLES

Column Index							
Row Index	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							
7							

The theory refers to “relations”

The implementation refers to “tables”

Each relation table has a name

The top row contains headings called **attributes**

An attribute corresponds to a column

Every other row (0 or more) is an **instance** of the relation and is defined by **a tuple** having components corresponding to the attributes

RELATION TABLES

ID	fname	surname	email	courseCode	currYear
16555666	Claire	Cox	c.cox555@nuigalway.ie	GY406	3
17444455	Marc	Bale	m.bale444@nuigalway.ie	GY350	2
17667788	Jack	Carr	j.carr667@nuigalway.ie	GY101	2
17987654	Marie	Berger	m.berger987@nuigalway.ie	GY101	2
17998877	Hugh	Flynn	h.flynn998@nuigalway.ie	GY350	2
18112233	Anna	Chikarovski	a.chikarovski122@nuigalway.ie	GY350	1
18123456	Donal	Nee	d.nee123@nuigalway.ie	GY101	1
18333222	Sadhbh	O'Malley	s.omalley333@nuigalway.ie	GY350	1
18654321	Sean	Lynch	s.lynch654@nuigalway.ie	GY101	1



Recall that mathematical relations do not contain duplicates:



- In relation tables no two tuples can be exactly the same (across all attributes).
- To ensure this completely, one or more special attributes are chosen (or added) which are called **primary key** attributes which **must** have unique values for each tuple.
- We use the convention (in writing) that attributes that form the primary key are underlined
- Graphically they are often represented with an image of a key.

RELATIONAL DBMS IN INDUSTRY

90% of industry applications use Relational DBMS or Relational DBMS with extensions.

The majority of industry applications require:

- Correctness
- Completeness
- Efficiency (Complex optimisation techniques and complex Indexing structures)

Relational DBMS provide this

RELATIONAL DBMS HAVE....

1. Design/**Structure View** where you can see structure of tables – names, data types and constraints
2. Datasheet/**Browse View** where you can see the database instance - data in the tables
3. Usually a results window
4. Usually a SQL editor (to write code)
5. And many other features

SAMPLE DESIGN/STRUCTURE VIEW

	#	Name	Type	Comments
<input type="checkbox"/>	1	id 	bigint(20)	
<input type="checkbox"/>	2	fname	varchar(50)	length 50
<input type="checkbox"/>	3	surname	varchar(50)	length 50
<input type="checkbox"/>	4	email	varchar(50)	length 50
<input type="checkbox"/>	5	courseCode	varchar(5)	length 5
<input type="checkbox"/>	6	currYear	int(11)	

Where we specify:

- Attribute (column) names
- Attribute (column) data types
- Primary key

SAMPLE DATASHEET/BROWSE VIEW

ID	fname	surname	email	courseCode	currYear	
16555666	Claire	Cox	c.cox555@nuigalway.ie	GY406	3	
17444455	Marc	Bale	m.bale444@nuigalway.ie	GY350	2	
17667788	Jack	Carr	j.carr667@nuigalway.ie	GY101	2	
17987654	Marie	Berger	m.berger987@nuigalway.ie	GY101	2	
17998877	Hugh	Flynn	h.flynn998@nuigalway.ie	GY350	2	
18112233	Anna	Chikarovski	a.chikarovski122@nuigalway.ie	GY350	1	
18123456	Donal	Nee	d.nee123@nuigalway.ie	GY101	1	
18333222	Sadhbh	O'Malley	s.omalley333@nuigalway.ie	GY350	1	
18654321	Sean	Lynch	s.lynch654@nuigalway.ie	GY101	1	

Where we enter the actual data

TABLE 1: addressbook

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

* See files on Blackboard for the sample tables used in lectures: ct102_2021.accdb and csv files

EXAMPLE 1: using table 1

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

Number of attributes?

Name of attributes?

Data type of attributes?

Number of rows?

Any duplicates?

Any potential duplicates?

ALTERNATIVE IF NOT USING DATABASE SYSTEM?

Easy to store the data from a single table in a text file and write a program, in programming language of choice, to open file and access data



The screenshot shows a Windows Notepad window titled "AddressBook.txt - Notepad". The window contains a list of 8 entries, each consisting of a name, address, town, county, country, phone number, and mobile number, separated by commas. The entries are:

- "Peter Smith",12,"Tudor Vale","Oranmore","Galway","Ireland","091888666","085454545"
- "Ali Byrne",31,"Station Road","Athenry","Galway","Ireland","091888444","085989811"
- "Cheryl Ainsley",131,"Cherry Gardens","Newcastle","Galway","Ireland","091232323","086123123"
- "Chris Nowak",,"Golf Road","Westport","Mayo","Ireland","098660012","086876543"
- "Ben Okoro",31,"Clare's Walk","Ennis","Clare","Ireland","065767676","087123456"
- "Gabe Jones",,"Dun Mor","Roundstone","Galway","Ireland","095333666","087232323"
- "Jane Doyle",,"Claremount","Claremorris","Mayo","Ireland","0949367821","087665544"

TABLE 2: appointments

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Class
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

EXAMPLE 2: USING TABLE 2 (appointments)

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Class
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

Number of attributes?

Name of attributes?

Data type of attributes?

Number of rows?

Any duplicates?

Any potential duplicates?

FUNDAMENTAL CONCEPT IN RELATIONAL DATABASES: PRIMARY KEY

It is very important that the primary key is unique and unambiguous and remains so even when new, yet unseen, data is added to a table.

Repetition in a primary key is ruled out theoretically and also not desirable in practical terms.

Often considerable effort is involved in the choosing, or creation, of a primary key

Examples of good primary keys

- PPS numbers or equivalent (unique within a country)
- Student IDs, Staff IDs (unique within an organisation)
- Bank account numbers (unique within a bank)
- Hospital chart numbers (unique within a hospital)
- Car registration numbers (unique within a country/region)

Others?

- Mobile phone numbers?
- Email addresses?
- Usernames?

CHOOSING A PRIMARY KEY? (1 OF 2)

In general want the **simplest** primary key possible:

- Not too long if possible – but length dependent on number of keys potentially required
- Chosen from existing attributes rather than having to add new one if possible
- Not too many attributes, one is best if possible
- Not too complex a data type, e.g. integers are easiest!

CHOOSING A PRIMARY KEY? (2 OF 2)

- Some existing attribute may be unique and can be chosen
- Some combination of existing attributes may be unique (in combination) and can be chosen (if not too many and if not too complex)
- Some new (“artificial”) attribute can be picked and added (e.g., autonumber datatype).

TABLE 1: Suitable primary key for addressbook table?

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh
Peter Smith	12	Tudor Vale	Oranmore	Galway	Ireland	091888666	085454545
Ali Byrne	31	Station Road	Athenry	Galway	Ireland	091888444	085989811
Cheryl Ainsley	131	Cherry Gardens	Newcastle	Galway	Ireland	091232323	086123123
Chris Nowak		Golf Road	Westport	Mayo	Ireland	098660012	086876543
Ben Okoro	31	Clare's Walk	Ennis	Clare	Ireland	065767676	087123456
Gabe Jones		Dun Mor	Roundstone	Galway	Ireland	095333666	087232323
Jane Doyle		Claremount	Claremorris	Mayo	Ireland	0949367821	087665544

TABLE 2: Suitable primary key for appointments table?

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Class
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

RECALL: DATABASE DEFINITION

One or more **tables**

- where a **table** is an ordered collection of **records**
- where a record consists of data

ONE OR MORE TABLES?

A relational database *could* consist of just one large table

For many purposes, this would be impractical and inefficient and would be difficult to update (i.e., add, modify or delete tuples or data).

The table would contain a great deal of **redundancy**

DEFINITION: Redundancy

Unnecessary **duplication** of data in a table as a result of data not being split into multiple tables

Duplication:

- If an attribute in a database has two identical values
- Data may be duplicated without being redundant
- Data is duplicated rather than redundant if when deleting or restructuring data in to multiple tables, information is lost

CONSEQUENCES OF REDUNDANCY

- Space is wasted
- Data can become inconsistent (data integrity is lost)
- Problems with update, insert and delete operations

Redundancy in appointments table?

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Class
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

NORMALISATION

All tables in a relational database must satisfy certain desirable properties

A hierarchy of “normal forms” exist that impose increasing restrictions on tables

These normal forms use “functional dependencies”

These normal forms are called:

- 1st, 2nd and 3rd normal forms
- Boyce-Codd (BCNF) normal form
- 4th and 5th normal forms

FUNCTIONAL DEPENDENCY

An attribute Y is functionally dependent on X, if knowing X can uniquely determine Y

e.g., if Y = name and X = studentID

The attribute *name* is functionally dependent on the attribute *studentID* as knowing a *studentID* can uniquely determine a *name*

Note: the reverse is **not** true.

Functional dependencies present in appointments table?:

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Class
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

REMOVING/REDUCING REDUNDANCY

Split data in to multiple tables according to functional dependencies

Important:

- No information should be lost
- Some attributes may exist more than once across multiple tables and this allows tables to be **linked** and cross-referenced (can be considered necessary duplication of data)

A better ordering of attributes appointments table? (Using multiple tables)

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate	Comments
1	Peter Murphy	1986	Prof Keogh	113	ENT	12/11/2021	
2	Ali Byrne	2001	Dr Lee	201	Gastro	23/11/2021	
3	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	25/01/2021	
4	Chris Nowak	1980	Prof Keogh	113	ENT	21/01/2022	
5	Cheryl Ainsley	1995	Dr Garvey	205	Dermatology	23/11/2021	
6	Jane Doyle	1988	Mr Gormley	101	ENT	30/11/2021	
7	Ben Okoro	1969	Mr Comer	107	Ophthalmology	20/01/2022	
8	Ali Byrne	2001	Mr Gormley	101	ENT	20/01/2022	
9	Gabe Jones	1998	Dr Garvey	205	Dermatology	01/02/2022	
*		0					

TABLE 3: School table

Choose an appropriate primary key (if possible)

Identify any redundancy in the table

Identify any functional dependencies (based on your knowledge of the domain)

Suggest a better ordering of attributes than that given (potentially in multiple tables)

ID	SName	Code	ModName	Lecturer	Location	Grade	ModCode	Yr
20343	A. Alabbad	GY101	Mathematics	G. Ellis	Arus De Brun	A	MA280	2
20343	A. Alabbad	GY101	Psychology	G. Molloy	Eng Building	A	PS414	2
21112	J. Bandewar	GY350	Mathematics	G. Ellis	Arus De Brun	B	MA160	1
21112	J. Bandewar	GY350	Electronics	J. Breslin	Eng Building	A	EE130	1
21222	J. Byrnes	GY350	Computer Systems	F. Glavin	IT Building	B	CT101	1
20178	M. Smyth	GY350	Computer Systems	I. Ullah	IT Building	C	CT213	2
20178	M. Smyth	GY350	Database Systems I	J. Griffith	IT Building	B	CT230	2
20178	M. Smyth	GY350	OO Programming	F. Glavin	IT Building	C	CT2109	2

School X

ID	SName	Code	ModName	Lecturer	Location	Grade	ModCode	Yr
20343	A. Alabbad	GY101	Mathematics	G. Ellis	Arus De Brun	A	MA280	2
20343	A. Alabbad	GY101	Psychology	G. Molloy	Eng Building	A	PS414	2
21112	J. Bandewar	GY350	Mathematics	G. Ellis	Arus De Brun	B	MA160	1
21112	J. Bandewar	GY350	Electronics	J. Breslin	Eng Building	A	EE130	1
21222	J. Byrnes	GY350	Computer Systems	F. Glavin	IT Building	B	CT101	1
20178	M. Smyth	GY350	Computer Systems	I. Ullah	IT Building	C	CT213	2
20178	M. Smyth	GY350	Database Systems I	J. Griffith	IT Building	B	CT230	2
20178	M. Smyth	GY350	OO Programming	F. Glavin	IT Building	C	CT2109	2

DATABASE LANGUAGES

The programming language for Relational Databases is called SQL - Structured Query Language

SQL is a standardised Query language across all relational DBMS (with some minor variations):

- First version SQL-89
- SQL-92 (SQL-2)
- SQL-99 (SQL-3)
- Recent standards include XML-related features

Standardised by American National Standards Institute (ANSI) and International Standards Organization (ISO)

SQL

- SQL is a **declarative language**
- It allows you specify the results you require ... not the order of the operations to retrieve those results
- In comparison, C, C++, Java, Python are considered **Imperative Languages** ... which facilitate computation by means of state changes, e.g., can specify

```
int a;  
a = 3;
```

SQL

Allows for specification of *queries*

Queries represent *information needs*

Queries can be run to produce results

Result might be:

- Output to user
- Modification of Data in Database
- CRUD operations: Create Read Update Delete

Read using SQL SELECT statement

Most important and often-used query is that of **selecting tuples from a table (or multiple tables) that satisfy some condition**

SELECT statement allows this

General format is:

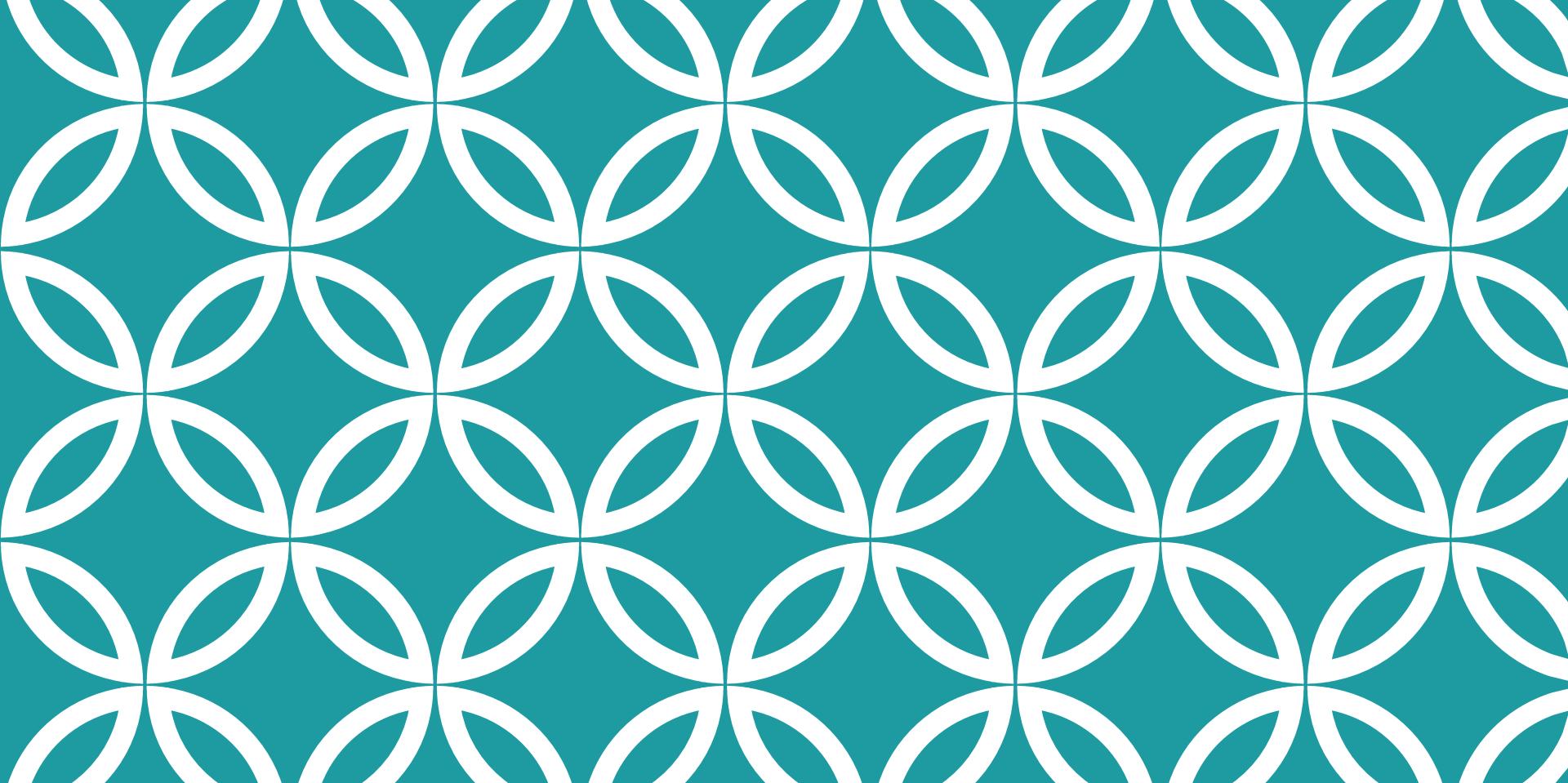
SELECT [DISTINCT] <attribute list>

FROM <table list>

WHERE <condition>

SUMMARY (PART 1)

- A database stores data in a structured format – having named columns (attributes) and their associated data type
- A primary key is a special attribute that has a unique value for each row of data entered to the table
- A database can have many tables
- Redundant data is often removed/reduced by considering functional dependencies and creating new tables
- A special programming language called SQL is used with relational database systems (and many other database systems use a language similar to SQL)
- We will cover SQL next week



DATABASE SYSTEMS: SQL

CT102:
Information
Systems

DATABASE LANGUAGES

The programming language for Relational Databases is called SQL - Structured Query Language

SQL is a standardised Query language across all relational DBMS (with some minor variations):

- First version SQL-89
- SQL-92 (SQL-2)
- SQL-99 (SQL-3)
- Recent standards include XML-related features

Standardised by American National Standards Institute (ANSI) and International Standards Organization (ISO)

SQL

- SQL is a **declarative language**
- It allows you specify the results you require ... not the order of the operations to retrieve those results
- In comparison, C, C++, Java, Python are considered **Imperative Languages** ... which facilitate computation by means of state changes, e.g., can specify

```
int a;  
a = 3;
```

SQL

Allows for specification of *queries*

Queries represent *information needs*

Queries can be run to produce results

Result might be:

- Output to user
- Modification of Data in Database
- CRUD operations: Create Read Update Delete

Read using SQL SELECT statement

Most important and often-used query is that of **selecting tuples from a table (or multiple tables) that satisfy some condition**

SELECT statement allows this

General format is:

SELECT [DISTINCT] <attribute list>

FROM <table list>

WHERE <condition>

Examples using addressbook table

appointments	AddressBook						
FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh

1 Using the original table 1, write a query to find the names and mobile phone numbers of all people in Galway.

```
SELECT FullName, MobPh  
FROM AddressBook  
WHERE county = 'Galway';
```

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh

2 Using the original table 1, write a query to find the name of the person with mobile phone number 087123456

SELECT

FROM

WHERE

Example using the appointments table

3 Using the appointments table, write a query to find the names and date of all appointments for the consultant “Dr Garvey”

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate
----	-------------	-----------	----------------	------	------------	---------

SELECT

FROM

WHERE

QUERYING ACROSS MULTIPLE TABLES

- A number of different approaches can be used if query needs to select data from multiple tables.
- The query becomes more complex. One approach is use two queries – an outer and a sub-query.
- If the subquery returns a single number then can connect the two with a simple mathematical operator such as $=$, \neq , $>$, $<$, etc.
- If the subquery returns a single string then can connect the two with a string comparison using an operator such as $=$, \neq

EXAMPLE 4:

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate
----	-------------	-----------	----------------	------	------------	---------

Assume you are given the following three tables:

patient (pID, pName, BirthYear)

consultant (cID, cName, room,
speciality)

appointments (ID, pID, cID, AptDate)

Find what room Ali Byrne should attend for the appointment on '23/11/2021'

```
patient(pid, pname, birthyear)
consultant(cid, cname, room, speciality)
appointments(id, pid, cid, aptdate)
```

```
SELECT room
FROM consultant
WHERE cID IN
(SELECT cID
FROM appointment
WHERE AptDate = #23/11/2021# AND pID =
(SELECT pID
FROM patient
WHERE pName = 'Ali Byrne')
);

```

What does the query look like using the original appointments table?

ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate
----	-------------	-----------	----------------	------	------------	---------

```
SELECT    room  
FROM      appointments  
WHERE     AptDate = #23/11/2021#  
          AND pName = 'Ali Byrne';
```

EXAMPLE 5: USING SCHOOL TABLE

Using the school table, write a query to find the names of all students with an “A” grade in Mathematics

ID	SName	Code	Yr	ModName	ModCod	Lecturer	Location	Grade
----	-------	------	----	---------	--------	----------	----------	-------

```
SELECT  
FROM  
WHERE
```

USING AGGREGATE FUNCTIONS

SQL supports a number of aggregate functions which can be used in the SELECT clause

Examples include:

- **SUM**, **AVG**, **MIN**, **MAX** applied to numeric fields
- **COUNT** returns the number of tuples/values specified in a query

EXAMPLE 6

Using the school table, write a query to find how many people received an “A” grade across all subjects

	appointments	AddressBook	School						
	ID	SName	Code	Yr	ModName	ModCod	Lecturer	Location	Grade

```
SELECT COUNT(Sname)
FROM
WHERE
```

EXAMPLE 7

Using the appointments table (and using a subquery) write a query to find the *youngest person* who has an appointment

appointments						
ID	PatientName	BirthYear	ConsultantName	Room	Speciality	AptDate

```
SELECT PatientName  
FROM appointments  
WHERE BirthYear =  
(SELECT  
FROM );
```

EXAMPLE 8: LOOKING AT 2 NEW TABLES:

employees(employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle)

offices(officeCode, city, phone, addressLine1, addressLine2, state, country, postalCode, territory)

LOOKING AT THE DATA TYPES

employees(employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle)

offices(officeCode, city, phone, addressLine1, addressLine2, state, country, postalCode, territory)

Column	Type
officeCode	varchar(10)
city	varchar(50)
phone	varchar(50)
addressLine1	varchar(50)
addressLine2	varchar(50) <i>NULL</i>
state	varchar(50) <i>NULL</i>
country	varchar(50)
postalCode	varchar(15)
territory	varchar(10)

Column	Type
employeeNumber	int(11)
lastName	varchar(50)
firstName	varchar(50)
extension	varchar(10)
email	varchar(100)
officeCode	varchar(10)
reportsTo	int(11) <i>NULL</i>
jobTitle	varchar(50)

EXAMPLE 8 QUESTIONS: Write SELECT statements to find the following answers:

- 8.1** Find all the countries where there are offices.
- 8.2** Find all the employees (their names) with job Title “Sales Rep”.
- 8.3** Find the cities in country “USA” where there are offices.
- 8.4** Find the email address of employee “Julie Firrelli”.
- 8.5** Find the postcode of the Paris office.

INSERT STATEMENT

The INSERT statement allows data to be inserted as part of a query (rather than via the graphical user interface (GUI))

General format is:

```
INSERT INTO table (<attribute list> )
```

```
VALUES (<value list>);
```

FullName	HseNum	Address1	Address2	County	Country	HomePh	MobPh

EXAMPLE 9

Add a new tuple to the AddressBook table for name ‘Ann Lawlor’ and house number (HseNum) 12

```
INSERT INTO AddressBook  
        (FullName, HseNum)  
VALUES      ('Ann Lawlor', 12);
```

Note: If primary key exists, must specify it for any insertion

UPDATE

Can modify one or more records

General format is:

UPDATE *table*

SET <attribute name> = <some value>

WHERE <condition>;

EXAMPLE 10

Update the house number of Peter Smith in the AddressBook Table to 90

```
UPDATE AddressBook  
SET HseNum = 90  
WHERE FullName = 'Peter Smith' ;
```

DELETE

The DELETE statement does not remove the table structure (e.g. attributes), only the data in the tables

General format:

DELETE *

FROM *table*

WHERE *condition*;

EXAMPLE 11

Delete appointment number 8 from the table appointments:

```
DELETE *
FROM    appointments
WHERE   id = 8;
```

YOU TRY ...

Example 12: for school table

Using `INSERT`, insert a new tuple into the school table for student “R. Sandip” with ID 181111 and Code GY350 and modCode ‘CT441’

`INSERT INTO`

`VALUES`

ID	SName	Code	ModName	Lecturer	Location	Grade	ModCode	Yr
20343	A. Alabbad	GY101	Mathematics	G. Ellis	Arus De Brun	A	MA280	2
20343	A. Alabbad	GY101	Psychology	G. Molloy	Eng Building	A	PS414	2
21112	J. Bandewar	GY350	Mathematics	G. Ellis	Arus De Brun	B	MA160	1
21112	J. Bandewar	GY350	Electronics	J. Breslin	Eng Building	A	EE130	1
21222	J. Byrnes	GY350	Computer Systems	F. Glavin	IT Building	B	CT101	1
20178	M. Smyth	GY350	Computer Systems	I. Ullah	IT Building	C	CT213	2
20178	M. Smyth	GY350	Database Systems I	J. Griffith	IT Building	B	CT230	2
20178	M. Smyth	GY350	OO Programming	F. Glavin	IT Building	C	CT2109	2

EXAMPLE 13: again with school table:

ID	SName	Code	ModName	Lecturer	Location	Grade	ModCode	Yr	
20343	A. Alabbad	GY101	Mathematics	G. Ellis	Arus De Brun	A	MA280	2	
20343	A. Alabbad	GY101	Psychology	G. Molloy	Eng Building	A	PS414	2	
21112	J. Bandewar	GY350	Mathematics	G. Ellis	Arus De Brun	B	MA160	1	
21112	J. Bandewar	GY350	Electronics	J. Breslin	Eng Building	A	EE130	1	
21222	J. Byrnes	GY350	Computer Systems	F. Glavin	IT Building	B	CT101	1	
20178	M. Smyth	GY350	Computer Systems	I. Ullah	IT Building	C	CT213	2	
20178	M. Smyth	GY350	Database Systems I	J. Griffith	IT Building	B	CT230	2	
20178	M. Smyth	GY350	OO Programming	F. Glavin	IT Building	C	CT2109	2	

Using UPDATE, change the grade for student with ID 21112 and modcode MA160 from “B” to “A”

Note: Boolean AND is written “AND” in SQL

UPDATE

SET

WHERE

Example 14 with school table

Using DELETE, delete student “A. Alabbad”, with ID 20343

DELETE

FROM

WHERE

ID	SName	Code	ModName	Lecturer	Location	Grade	ModCode	Yr
20343	A. Alabbad	GY101	Mathematics	G. Ellis	Arus De Brun	A	MA280	2
20343	A. Alabbad	GY101	Psychology	G. Molloy	Eng Building	A	PS414	2
21112	J. Bandewar	GY350	Mathematics	G. Ellis	Arus De Brun	B	MA160	1
21112	J. Bandewar	GY350	Electronics	J. Breslin	Eng Building	A	EE130	1
21222	J. Byrnes	GY350	Computer Systems	F. Glavin	IT Building	B	CT101	1
20178	M. Smyth	GY350	Computer Systems	I. Ullah	IT Building	C	CT213	2
20178	M. Smyth	GY350	Database Systems I	J. Griffith	IT Building	B	CT230	2
20178	M. Smyth	GY350	OO Programming	F. Glavin	IT Building	C	CT2109	2

DATABASE SYSTEM SUMMARY

A database is a set of tables, where a table is an ordered collection of **records**, where a record consists of a primary key and data.

A database requires some data access method in order to query and modify data

Important terms: table, attribute, tuple, instance, primary key

Redundancy and *idea of functional dependencies*

SQL SELECT statement **on 1 table only**

SQL INSERT, UPDATE, DELETE**on 1 table only**

TOPIC: RECOMMENDER SYSTEMS



CT102
Information
Systems

DEFINITION: RECOMMENDER SYSTEM



A recommender system provides suggestions for **items** to a **user** to support **decision-making processes**, such as:

- what items to buy: computers, phones, cameras, appliances, cars, etc.
- what music to listen to or what videos to watch
- what books or news articles to read
- what films to watch
- where to stay; where to eat
- who to connect to or link to

Generally the systems use the idea of **preferences** and/or **ratings** to make recommendations or use the existing **social links** between people.

PREFERENCES AND RATINGS



Preference: a greater liking for one alternative (or thing) over another or other thing (Oxford English Dictionary), e.g. choosing to watch one particular video over others in a list

Rating: a measurement of how good or liked something or someone is, e.g. likes, stars, actual rating value.

Like

Comment

Share

76

499

672



4.5 out of 5 stars

1,354,560 views

9,532

169

Characteristics of preferences and ratings



Preference data and **rating** data is viewed as an *indicator of user's likes and tastes*

Can be gathered **implicitly** or **explicitly**

Explicitly gathered using some meaningful icon (heart, thumbs up, stars, number, etc.)

Implicitly gathered based on a person's actions (viewing, listening, etc.)

WHAT DO WE MEAN BY IMPLICIT AND EXPLICIT PREFERENCE?

Preference questions:

Do I like the film “Dune”

Would I like the film
“Dune”

Do I like the book “Dune”

Explicit Preference: ?

Implicit Preference: ?



EXPLICIT RATINGS AND PREFERENCES

System will ask user to explicitly indicate “like” or “rate” or to give a value for one or more items.

Usually a numeric representation of the rating is stored if this is not directly provided by a person (i.e. a heart or star is stored as a number)



IMPLICIT RATINGS AND PREFERENCES



Try infer user's rating (opinion) based on actions taken by the user.

e.g.,

- Watching video and sharing videos
- Number of times listening to a mp3 and/or adding it to play list
- Posting a comment or liking something
- Purchasing something
- Time spent reading an item
- Following a new account or user

Usually numeric representations of preferences are stored and these may be weighted to indicate importance and/or time

SCORES

Usually in a range:

- Binary: 0, 1 (0 dislike; 1 like)
- -1, 0, 1 (-1 dislike; 0 neutral; 1 like)
- 1, 2, 3, 4, 5 (1 strong dislike; 5 strong like)
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 (1 strong dislike; 10 strong like)

Notes:

- Larger scales don't necessarily give better indicator of a person's likes and tastes
- People tend to user different "portions" of the scale

Numeric rating and preference data can be viewed as a matrix ...

PREFERENCES VS RATINGS VS REVIEWS?

Often a combination or all types of feedback are used

- Reviews have content (text) and may be more nuanced and require different techniques to process which may not be directly used for recommendation.
- Nowadays the idea of preferences and ratings are often combined
- Implicit ratings and preferences are generally more “noisy” than explicit ratings and preferences but are easier to obtain, i.e. are a by-product of using a particular app and require no extra effort by a person.
- We often don’t bother giving explicit ratings and reviews when asked (e.g., after purchasing a product)

MAIN DATA USED IN RECOMMENDATION SYSTEMS



- Unique ID for each users
- Unique ID for each items (whatever that might be, song, album, book, video, etc.)
- Ratings/Preference numbers (aggregated) for users for items
- Content of items (generally represented by text even if the item is non-text, e.g. video, music, etc.)
- Content associated with user (textual list of preferences, mood, location, age – provided by user or gathered by system based on implicit actions)

PROFILES (USER AND ITEM CHARACTERISTICS)



- The textual data associated with a user as well as the ratings and preference scores of a user for a set of items can be seen as representing the *user profile* and is unique for each user.
- The textual data associated with items, and the preference and rating scores an item receives from a set of users can be seen as representing the *item profile* and is unique for each item.

PROFILES

Can be stored in a database (structured) or free-form (unstructured text)

For example:

- For videos could store the video details, creator details, captions, hashtags, tune used, etc.
- For music, could store artist details, band details, genre, year, etc.
- For books, could store title, author, genre, abstract or summary, etc.

It is important to maintain the history of what a person has already ‘consumed’ as do not want to recommend something the person has already seen/liked/interacted with.

USING THIS DATA?

Can be used for personalised and non-personalised **recommendations**

WHAT APPROACHES ARE USED?

- Database
- Statistical (correlation)
- Matrix factorization
- Machine Learning

NON-PERSONALISED RECOMMENDATIONS



- Lists of items which can be generated for everyone.
- Often based on popularity, newness and velocity.
- **Popularity:** number of views, shares, likes, hashtag use, etc.
- **Newness:** recent content generally ranked higher.
- **Velocity:** The rate of the popularity growth (e.g., over minutes, hours, days).
- People often want to see/hear/know about what everyone else is talking about.
- Also useful for new users who have not given any indication of preferences yet and to offer diversity to users (something different to their personalised recommendations)

Trending videos

- Adele - Easy On Me (Official Video)
Adele 72M views • 3 days ago
Official Video for "Easy On Me" by Adele. Shop the "Adele" "Easy On Me" here: <http://Adele.Ink.to/EOM> Amazon Music: [ht...](#)
- THE BATMAN – Main Trailer
Warner Bros. Pictures 18M views • 1 day ago
It's not just a call... It's a warning. From Warner Bros. Pictures Robert Pattinson in the dual role of Gotham City's vigilante de...
- "Sore loser! An idiot!" Tyson Fury reveals w/ & Deontay Wilder after huge win
BT Sport Boxing 6.9M views • 1 week ago
Tyson Fury speaks to Steve Bunce and tells him what his work fight. Subscribe to BT Sport Boxing on YouTube [http://www...](#)

15:26 Mon 18 Oct

N Series Films Recently Added

Popular on Netflix

- ALICE IN BORDERLAND
- MY NAME

Trending Now

- YOU
- modernfamily

IMPLEMENTATION

N.B.

Very easy to implement and update.

Keep an ordered list of the content IDs based on the weighting of popularity + newness + velocity

The popular measure should be a **strong indicator** such as views, plays, likes, hashtags, etc.

These lists might be kept for different genres and displayed per genre, as well as the overall most popular across all genres being displayed for everyone.

Search Twitter

What's happening

Sports - LIVE
Steven Gerrard announced as new Aston Villa head coach
Trending with Gerrard

#AldiAmazingChristmas 🎄
Will Ebanana Scrooge discover his Christmas spirit?
Promoted by Aldi Ireland

Trending in Ireland
NPNET
2,014 Tweets

Trending in Ireland
#liveline

Business & finance - Last night
Elon Musk sells \$1.1 billion of Tesla stock

Show more

Wattpad Stories

Soup
Trending hashtag

Tomato Soup

Rocktober
Trending hashtag

Play along to In The Name Of Tragedy with Phil Campbell

Fitness
Trending hashtag

Want Forearms Like Batman? Do This

16 year old me thinking I could never build muscle ...

Home Discover

PERSONALISED RECOMMENDATIONS



Use **User profile** and **Item profile** information as an **indicator** of user's likes/tastes/preferences.

Recommendation task can be stated as: for some **active** user A recommend items not yet liked/seen by user A based on:

- I. the preferences of user A (Content-based recommendation).
- II. the preferences of other users with whom user A shares some preferences in common (Collaborative-based recommendation).

i) The preferences of a user: Content-based recommendation



Content-based recommendation recommends items based on items users have indicated a preference for in the past where the similarity between items is based on the **content** of those items.

The recommendation process therefore consists in matching the attributes of the user profile against the attributes of a content object.

More on Attributes

- The attributes used are dependent on the domain and ideally are well-defined (structured/using tags) so that meaningful matching can be performed using a query language.
i.e., so that a preference indicator can be associated with a genre, year, style, playlist, age category, creator name, etc.
- In the absence of the attributes being well defined (e.g., description) the main **descriptive terms** from the content are associated with the user profile and a more generic matching approach is used.
- In both cases, a higher weighting would be given to more recent preference indicators.
- Larger apps, like YouTube, Netflix, Amazon, etc., would use a combination of approaches.

CONTENT BASED RECOMMENDATION APPROACHES



1. Database approach (Relational or Semantic Web).
2. Can represent attributes as weighted terms in vectors and use Euclidean dot product to get recommendations.
3. Machine learning approaches (particularly deep learning approaches).

Going from simple and quick matching to more complex and time intensive matching

EXAMPLE: USING THE VECTOR APPROACH

(Taken from: buildingrecommenders.wordpress.com)



Given: the titles of six books and binary user preference data (based on ratings or purchases)

Assume: the active user A has already purchased the book “Introduction to Recommender Systems” and this is stored as positive indicator of the user’s preferences for the book

Task: use a content approach, with only titles available, to find which of the other 5 books user A should be recommended

A vertical list of six book titles. To the left of the list is a small thumbnail image of the books. The first book, "Introduction to Recommender Systems", is circled in red at the top of the list.

	Introduction to Recommender Systems
	Machine learning Paradigms
	Social Network-based Recommender Systems
	Learning Spark
	Recommender Systems Handbook
	Recommender Systems and the Social Web

CONTENT BASED APPROACH

- Assume we only have book titles available to use (attribute = title)
- In this scenario, a vector based approach and Euclidean dot product would work best where:
- Query is title of book user A has indicated a preference for (“Introduction to Recommender Systems”)
- This will be compared to all other books to find those most similar.

	Introduction to Recommender Systems
	Machine learning Paradigms
	Social Network-based Recommender Systems
	Learning Spark
	Recommender Systems Handbook
	Recommender Systems and the Social Web

STEPS:



Introduction to Recommender Systems

Machine learning Paradigms

Social Network-based Recommender Systems

Learning Spark

Recommender Systems Handbook

Recommender Systems and the Social Web

- Remove stop words from each title (to, and, the)
- As typically book titles (or other titles) will not have repeated words the weighting used will be Boolean (1 = present, 0 = absent).
- Represent each word in a vector with respect to the weighting.
- Start calculating similarity.

VECTOR REPRESENTATION OF BOOKS

	Introduction to Recommender Systems
	Machine learning Paradigms
	Social Network-based Recommender Systems
	Learning Spark
	Recommender Systems Handbook
	Recommender Systems and the Social Web



introduction	1				
recommender	1	1		1	1
systems	1	1		1	1
machine		1			
learning		1	1		
paradigms		1			
social			1		1
network-based			1		
spark				1	
handbook				1	
web					1

SIMILARITY:



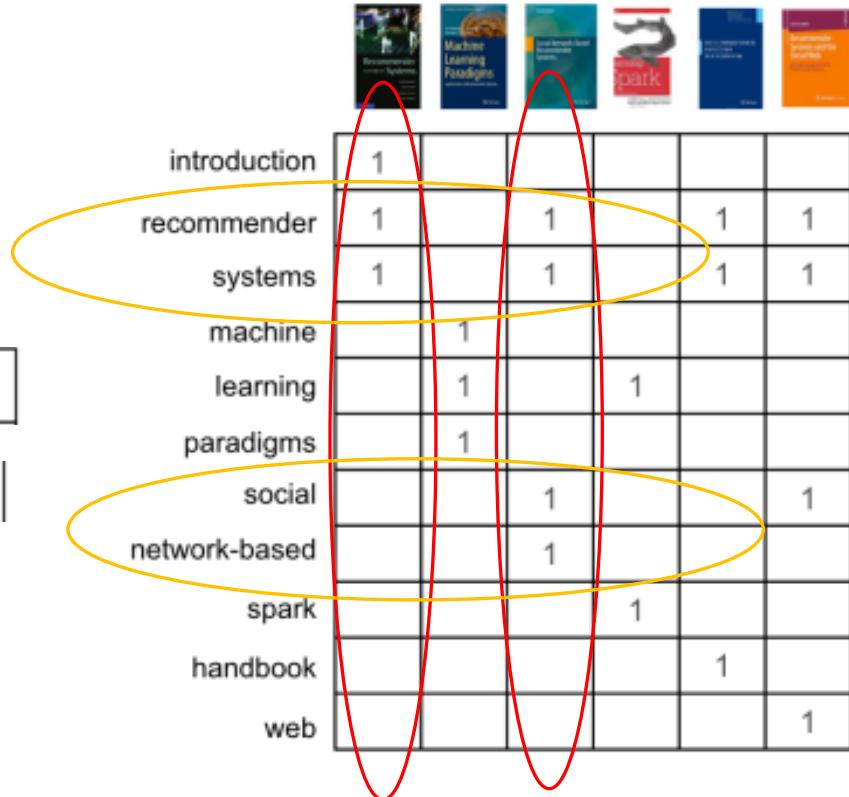
Introduction to Recommender Systems



Social Network-based Recommender Systems

<1, 1, 1, 0, 0, 0, 0, 0, 0, 0>

<0, 1, 1, 0, 0, 0, 1, 1, 0, 0>



$$\frac{2}{(\sqrt{3} \times \sqrt{4})} = 0.5773502691896258$$

= 0.58



SIMILARITY:



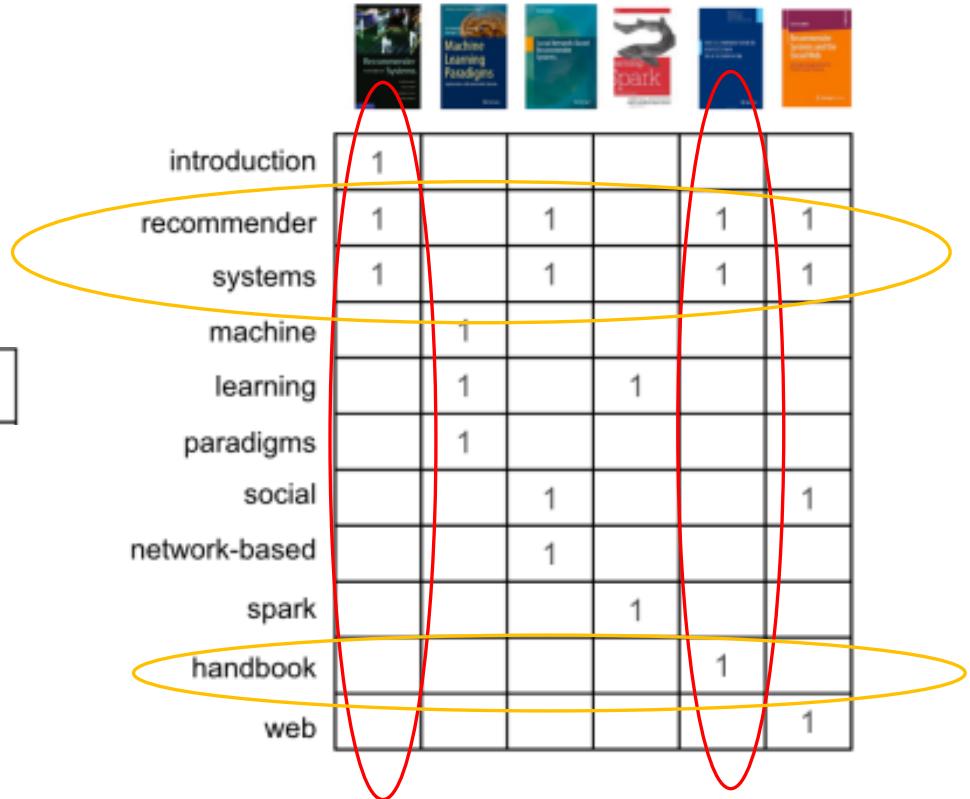
Introduction to Recommender Systems



Recommender Systems Handbook

<1, 1, 1, 0, 0, 0, 0, 0, 0, 0>

<0, 1, 1, 0, 0, 0, 0, 0, 0, 1>



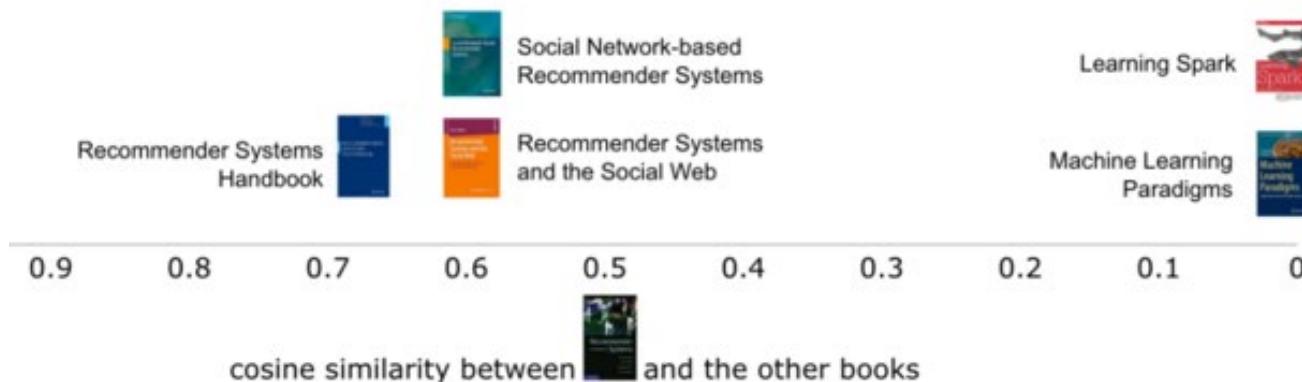
$$\frac{2}{(\sqrt{3} \times \sqrt{3})} = \frac{2}{3} = 0.6666666666666667$$

= 0.67



USING COSINE SIMILARITY TO FIND SIMILARITY OF BOOKS TO ACTIVE USER'S BOOK:

Similarity of all books to “Introduction to Recommender Systems”:



CAN PRE-CALCULATE THE SIMILARITY OF ALL BOOKS TO EACH OTHER:



HOW WOULD THIS WORK MORE GENERALLY?

This would be a good general approach for free-text (unstructured) attributes such as “description”.

In reality, for books other than academic books, attributes such as genre and author would be more useful for matching and would be much easier to implement with a database approach.

It is important to note though that for either approach the similarity of items to each other can be pre-calculated thus when/if a user indicates a preference for an item those items which are similar can be retrieved very, very quickly.

ADVANTAGES AND DISADVANTAGES OF CONTENT BASED APPROACH



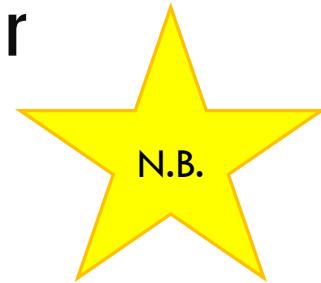
- No cold start problem (don't need many past user or item preferences)
- No popularity bias – can recommend rare items as long as they match the content of an item the user has indicated a preference for.
- Attributes can be weighted to keep user profiles current and fresh.
- Can provide an **explanation** of the recommendation ("because you watched this ... ")
- Item content (attributes) need to be machine readable and meaningful (e.g., genres, authors, text title, abstract text, etc.)
- "Stereotyping" possible – only get items similar to what you have already liked.
- Serendipity difficult – small chance of getting something unexpected or outside of your "filter bubble" (but un-personalised recommendations can help here)

SUMMARY OF THESE STRENGTHS AND WEAKNESSES FROM A RECOMMENDATION PERSPECTIVE ...

Content based approaches are good at **Exploitation**- items similar to those for which the user has already expressed a preference are recommended ... but the approaches are not as good at **Exploration** - items with no content in common with what the user has previously indicated a preference for will not be recommended.

Approach (ii) the preferences of other users with whom user *A* shares some preferences in common

Collaborative-based recommendation



Does not use any content data.

Based on the intuition that “user preferences are correlated”.

Tries to mimic “word of mouth” recommendation that often happens between people.

Uses user **preference** data to form similarity between **users** and between **items**.

EXAMPLE

Adapted from one of the earliest papers on collaborative filtering by Resnick et al., 1994

Given: Ratings by 4 users for 6 movies (where movies are represented by IDs). Ratings are in the range 1-5 (1=“strong dislike”; 5 = “strong like”)

<i>Movie #</i>	<i>Ken</i>	<i>Lee</i>	<i>Meg</i>	<i>Nan</i>
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6		2	5	

Sample Goals:

Which users have similar tastes?

Find whether Ken is interested in (likes) movie 6

Will Nan like movie 6?

ADVANTAGES OVER CONTENT FILTERING:

- Support for recommendation of items where content cannot be analysed easily in an automated manner
- Ability to take issues of taste into account that may be difficult to represent with the content-based approach
- Serendipitous recommendations possible – can recommend items without the user needing to have seen items similar to the item previously or without the user needing to state an explicit preference for that type of item (e.g. genre of music, etc.)

A large yellow five-pointed star with a thin black outline, positioned in the bottom right corner of the slide.

N.B.

APPLICATIONS

- The approach has been successful in a number of domains
- Especially when a person can easily consume content (e.g. Netflix, Amazon Prime, Spotify etc.) but may not want to invest time and effort in explicitly specifying their preferences.
- Most well-known examples involve recommending music, videos, books, streaming, e-commerce and social media domains.
- Often combined with other approaches (content and popularity) as they compliment each other in terms of what they each offer.

MAIN DATA USED IN COLLABORATIVE FILTERING APPROACH:



- Users (represented by an ID)
- Items (represented by an ID)
- Ratings/Preference scores of users for items

MATRIX DATA REPRESENTATION

In this example,

- Values are in the range [1-5]
- 0 indicates no value
- Rows represent users
- Columns represent items
- Generally, unless storing a very small amount of data, would not store in this format (would only store non-zero data)

```
0 0 0 0 0 0 0 0 0 0 0 0 0 4 3 3  
0 0 4 3 5 0 3 4 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0  
0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 4 0 5 0 0 0 0 0 0 0 0 2 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0  
0 0 4 3 4 0 0 0 0 0 4 4 0 0 0 0 0  
4 0 0 2 5 0 0 0 4 3 0 0 0 0 0 0 0  
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

FEATURES OF THE DATA



Typically, as can be seen from the matrix representation, there are many 0 entries.

This is referred to as matrix *sparsity* i.e., there are many items for which we do not have a value (no preference or rating from the user).

For many real sets of data on average only 1% of items will have rating/preference scores.

This makes sense given the number of items which exist but it can make it difficult to find similar items and users if there is no “overlap” between user/item preferences.

STORAGE OF SAMPLE DATA



Usually stored as a triple:

- userID, itemID, rating
- e.g.,

196, 242, 3

186, 302, 3

22, 377, 1

244, 51, 2

166, 346, 1

298, 474, 4

115, 265, 2

253, 465, 5

movielens_sa...	
File	Edit
Format	View
Help	
196,242,3	
186,302,3	
22,377,1	
244,51,2	
166,346,1	
298,474,4	
115,265,2	
253,465,5	
305,451,3	
6,86,3	
62,257,2	
286,1014,5	
200,222,5	
210,40,3	
224,29,3	
303,785,3	
122,387,5	
194,274,2	
291,1042,4	
234,1184,2	
119,392,4	
167,486,4	
299,144,4	
291,118,2	
308,1,4	
95,546,2	
38,95,5	
102,768,2	
62 277 1	

COLLABORATIVE FILTERING APPROACHES

- Numerous approaches have been researched and tested.
- Modern approaches often use some form of machine learning technique.
- A common technique which is easy to understand is a statistical user-user or item-item correlation, neighbour-based approach which finds groups of similar users or similar items.
- For practical applications, it is more useful to find and store similar items.

STEPS IN A PEARSON CORRELATION NEIGHBOUR-BASED APPROACH



1. **Calculate user-user or item-item similarity:** Find how similar each user/item is to every other user/item. Many approaches possible – mostly using maths techniques, IR techniques (vector similarity) or machine learning techniques. We will look at a popular and simple approach using the [Pearson correlation formula](#).
2. **Select Neighbourhood:** Form groups or neighbourhoods of users/items who are similar based on the calculations in part 1.
3. **Generate Recommendation:** In each group, can make recommendations based on other similar users/items

PEARSON CORRELATION: Weighted average of deviations from the neighbours' mean is calculated

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2}}$$



where for m items:

$r_{a,i}$ is rating of user a for item i

\bar{r}_a is the average rating given by user a

$r_{u,i}$ is rating of user u for item i

\bar{r}_u is the average rating given by user u

Notes: The result is in the range [-1 ,1].

The correlation can be computed only if there are common items rated by both the users

NOTE:

Alternative formulae also exist:

$$r = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{n}\right) \left(\sum Y^2 - \frac{(\sum Y)^2}{n}\right)}}$$

- * However the previous version is easier to remember as it looks more similar to the cosine similarity formula

2. SELECT NEIGHBOURHOOD:



Some approaches:

Correlation thresholding: where all users with similarity above a certain threshold are selected. Threshold usually:

- 0 if only want to look at positive correlations
- Usually slightly over 0, e.g. 0.1

Best-N correlations: where the n neighbours who have the highest similarity are chosen

SIZE OF NEIGHBOURHOOD?

Is there a best number of neighbours to pick?

If a large number of neighbours are picked:

- Potentially get *low precision* predictions

If a small number of neighbours are picked?

- Potentially get few or no predictions

3. GENERATE RECOMMENDATION:

For some user (the active user):

- make recommendations based on what the **neighbours** have rated but the active user has not rated
- often the **weighted average** of neighbor's ratings are used

3. GENERATE RECOMMENDATION: To give a prediction, P , for an item i for an active user a use weighted average approach:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u=1}^n w_{a,u}}$$



- $\underline{r_a}$ is the average rating given by active user a
- $r_{u,i}$ is rating of user u for item i
- $\underline{r_u}$ is the average rating given by neighbour u
- $\underline{w_{a,u}}$ is the similarity between user u and a

EXAMPLE 1 AGAIN



<i>Movie#</i>	<i>Ken</i>	<i>Lee</i>	<i>Meg</i>	<i>Nan</i>
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	

Task:

Find recommendation for
movie 6 for Ken

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2}}$$

STEPS:



1. Use a Pearson Correlation approach to find similar users
2. Include all users with positive correlations as neighbours (similarity > 0)
3. Use a weighted average approach for recommendation

1. FINDING SIMILAR USERS

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2}}$$

First calculate averages for each person

<i>Movie#</i>	<i>Ken</i>	<i>Lee</i>	<i>Meg</i>	<i>Nan</i>
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	

$$\text{Avg(Ken)} = (1+5+2+4)/4 = 3$$

$$\text{Avg(Lee)} = (4+2+5+1+2)/5 = 2.8$$

$$\text{Avg(Meg)} = (2+4+3+5)/4 = 3.5$$

$$\text{Avg(Nan)} = (2+4+5+1)/4 = 3$$

1. FINDING SIMILAR USERS

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2}}$$

Now find correlations

Movie#	Ken	Lee	Meg	Nan
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	

$$\begin{aligned}\text{Avg(Ken)} &= 3 \\ \text{Avg(Lee)} &= 2.8 \\ \text{Avg(Meg)} &= 3.5 \\ \text{Avg(Nan)} &= 3\end{aligned}$$

corr(Ken, Lee) :
Top Line:

$$(1-3) \times (4-2.8) + (5-3) \times (2-2.8) + (2-3) \times (5-2.8) + (4-3) \times (1-2.8) =$$

Ken denominator:

$$\sqrt{(1-3)^2 + (5-3)^2 + (2-3)^2 + (4-3)^2}$$

Lee denominator:

$$\sqrt{(4-2.8)^2 + (2-2.8)^2 + (5-2.8)^2 + (1-2.8)^2}$$

$$\text{corr(Ken, Lee)} = -0.79$$

PEARSON CORRELATION SIMILARITIES

	Ken	Lee	Meg	Nan
Ken		-0.79	0.89	0
Lee			-0.95	0.59
Meg				0.89
Nan				



3. GENERATE RECOMMENDATION:



$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u=1}^n w_{a,u}}$$

- What we want to know is if this value is ≤ 3 or > 3
- It will only be recommended if it is > 3
- $P_{ken, movie6} =$
$$3 + \frac{(5 - 3.5) \times 0.89}{0.89}$$
- $P_{ken, movie6} = 4.5$

Movie#	Ken	Lee	Meg	Nan
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	

CONSIDER USING AN ITEM-ITEM NEIGHBOUR BASED APPROACH

e.g. Calculate the similarity of movies 2 and 6 using the Pearson correlation formula

avg of movie 2 = 3.75; avg of movie 6 = 3.5

Movie 2 ratings: 5, 2, 4, 4

Movie 6 ratings: 0, 2, 5, 0



$$\frac{((2 - 3.75) \times (2 - 3.5) + (4 - 3.75) \times (5 - 3.5))}{\left(\sqrt{(2 - 3.75)^2 + (4 - 3.75)^2} \times \sqrt{(2 - 3.5)^2 + (5 - 3.5)^2} \right)} = \frac{4}{5} = 0.8$$

EXAMPLE 2



We are given the following information about 4 users and we wish to make a recommendation for the movie “Black Widow” for Sam who has an average rating of 4:

Joe has rated “Black Widow” 4 and has an average rating of 3.5

Ana has rated “Black Widow” 5 and has an average rating of 3

Ali has rated “Black Widow” 4 and has an average rating of 4.5

The correlations between the users have been calculated as follows:

$$\text{corr}(\text{Sam}, \text{Joe}) = .79$$

$$\text{corr}(\text{Sam}, \text{Ana}) = .57$$

$$\text{corr}(\text{Sam}, \text{Ali}) = 0$$

HYBRID RECOMMENDER APPROACHES

- Combines more than one approach to make recommendations.
- Approaches combined depend on the data that is available.
- Usually combine popularity, collaborative and content approaches – where each approach can outweigh the disadvantages of the other.
- Recent approaches are integrating data from LOD such as DBpedia and MusicBrainz.org etc.



NETFLIX incorporated one of the earliest successful recommendation systems and is now just one of many streaming apps which offer different types of recommendations in order to maintain and grow its customer base. Multiple profiles can exist and profiles are augmented each time a new film or series is watched.

These recommendations can be based on:

- Popularity/trending, e.g., number of views.
- Similarity to other films, series that have already been viewed by a user.
- User similarity and item similarity based on people who have watched them.



TikTok uses personalised recommendation techniques for the **#ForYou** feed feature based on each user's interactions with the content viewed.

Initially users are asked to provide explicit indicators of their preferences – i.e., categories or genre of content they are interested in, fashion, sport, cooking, etc.

Once a user starts viewing videos and comments, shares or replays a video this is seen as a **positive indicator** of the user's preferences and is used to display similar content to the user.

The better TikTok's recommendation algorithm is, the longer people will stay on the app, and the more likely they are to return sooner to see new content.



- Instagram feeds provide personalised content to users based on the user's interaction with previous content on Instagram.
- Machine learning techniques are used to generate recommendations based on:
 - User preferences.
 - Recent content (based on time) and thus it is more likely you won't have seen this content – especially it has been uploaded since you last checked Instagram.
 - Social: Number of accounts user follows and the accounts that the user interacts most with.
 - Frequency and usage: How often you open the app and how long you stay on the app so that you get fresh, relevant content every time (easier if you only check once a day versus once an hour!)



- YouTube provides a personalised “Suggested Videos” feature as well as automatically playing videos when you have finished watching the current video. Also offers “Trending” page for popular (non-personalised) suggestions.
- The main data used for recommendation is:
 - Title, description, and keywords of videos.
 - User Preference or Engagement measure (watch time, likes, comments, etc.) when they watch videos.
 - Since 2016 YouTube has been using machine learning techniques (deep learning) for its recommendations.

SUMMARY

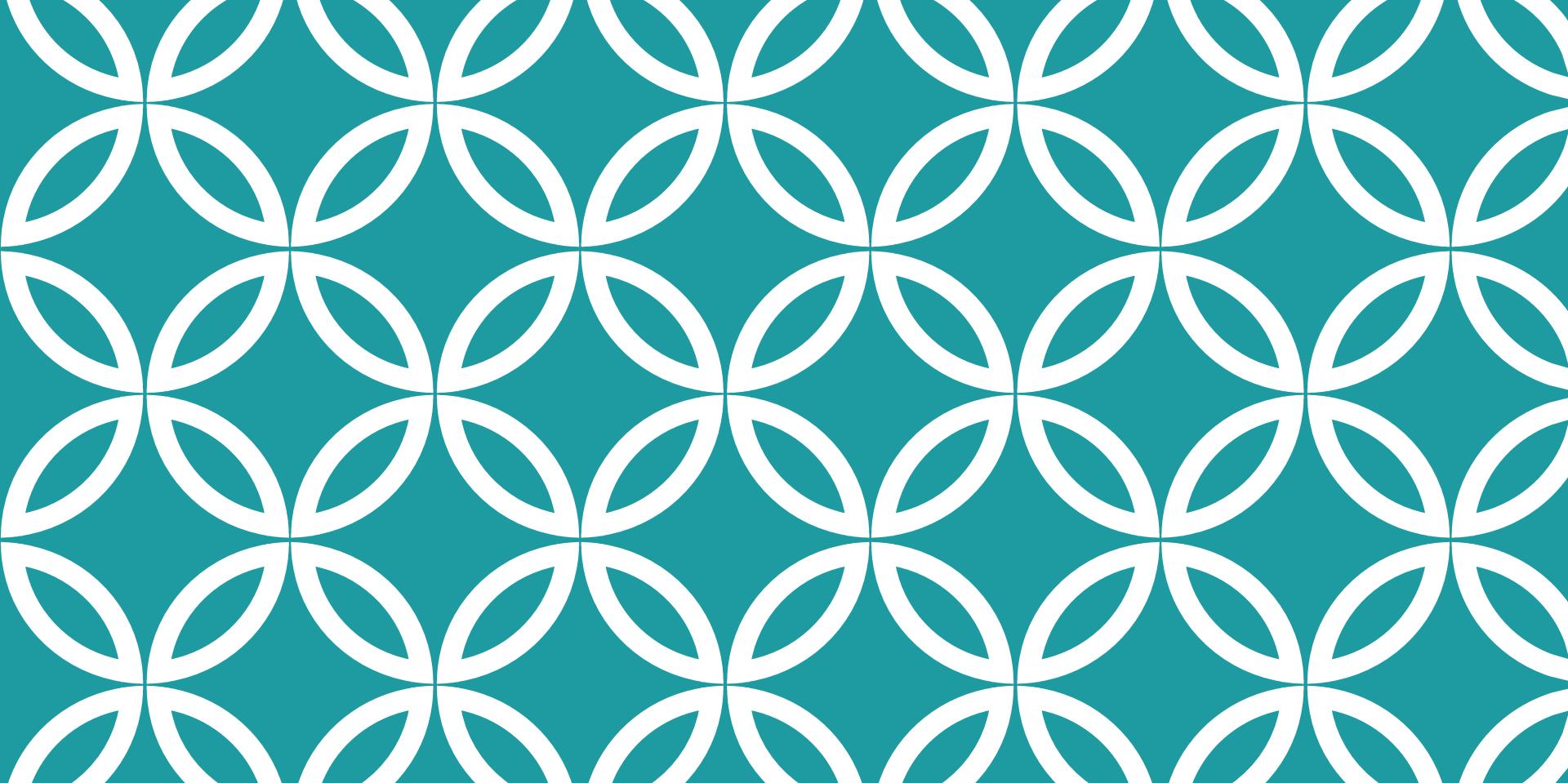
Filtering and Recommendation are a large area of research and a huge application area on the web in particular with most systems offering some form of recommendation.

Many approaches used using different data - content, rating - statistical and machine learning – we concentrate on one approach – a nearest neighbour approach.

In general, amounts of data collected and used are huge.

WHAT'S IMPORTANT TO KNOW

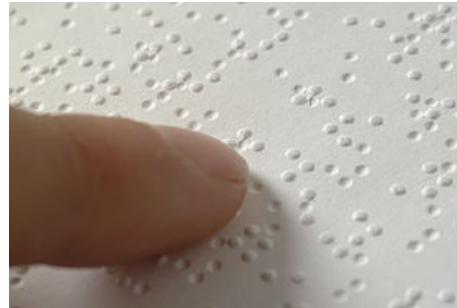
- difference between implicit and explicit ratings
- difference between non-personalised and personalised recommendations
- difference between content-based and collaborative-based recommendations - what data they use and how they differ in what they can recommend
- how a content-based approach (using vectors and Euclidean dot product) works
- how a collaborative-based approach (using Pearson correlation and weighted average) works



INTRODUCTION TO **DATA COMPRESSION**

CT102:
Information
Systems

COMPRESSION



Reduces the space a file or message occupies

Specifically:

- encoding information using fewer bits than the original representation

A	B	C	D
• -	- ...	- - .	- ...
E	F	G	H
•	• - - •	- - •	• ...
I	J	K	L
..	• - - -	- - .	- ...
M	N	O	P
- -	- .	- - -	• - - .
Q	R	S	T
- - -	• - -	-
U	V	W	X
• - -	• ... -	- - -	- ... -
Y		Z	
- - -		- - -	

WHY?

To **save space** when storing

To **save time/bandwidth** when transmitting

Still needed?

Moore's law:

- Number of transistors on a chip doubles every 18-24 months ...

Parkinson's Law:

- Data expands to fill the space available for storage/transmission

HOW?

Many techniques are based on the fact that most files have **redundancy**. Examples:

- **Text:** ths sntnc cn b rd rthr qckly by mst ppl



- **Images:** large areas of same colour



- **Videos:** frames that are very similar to last

SOME EXAMPLES

Data: Gzip, Boa, Pkzip, Brotli

Images:

- **.gif** (graphics interchange format). Lossless
- **.jpg** (joint photographic experts group). Lossy. Full-colour or gray-scale digital images of "natural", real-world scenes
- **.png** ... gif like ... not as common as gif or jpeg

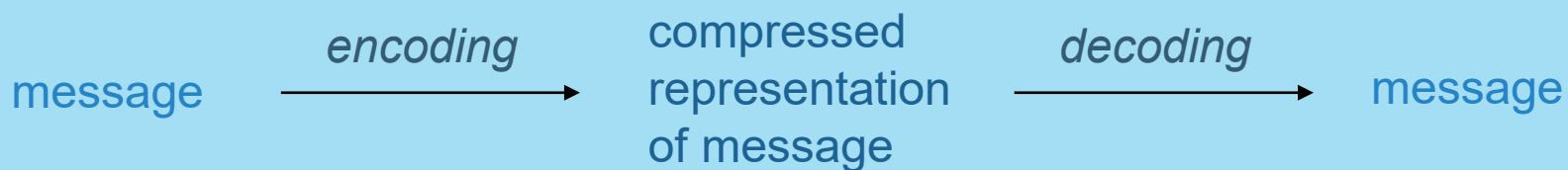
Sound: MP3

Video: MPEG, DivX, HDTV

TWO COMPONENTS: Encoding and Decoding

Goal:

- The encoded message (compressed representation of the message) has fewer bits than the original message
- The decoded message is the same, or approximately the same as the original message



A FREE LUNCH? NO!

To be used **must** undergo the opposite process of decompression

An example of a space-time complexity trade-off (common in computing):

- i.e., storage and transmission time gains versus execution (CPU) time for encoding and decoding

HIGH LEVEL CATEGORISATION OF COMPRESSION ALGORITHMS

- Lossy
- Lossless
- Hybrid

LOSSY COMPRESSION ALGORITHMS

Loose some information ... in general not noticeable to human eye.

Cannot be used for text files or images that need to be closely analysed (e.g., medical images).

LOSSLESS COMPRESSION ALGORITHMS

No loss of information

Can be used for text files and any image files
that need to be closely analysed

Is there a lossless algorithm that can compress all file types?

No need to assume some **bias** in the data (message) and exploit this bias to reduce the size of the file

QUALITY OF COMPRESSION APPROACHES

Lossless:

Time taken to encode

Time taken to decode (e.g., when streaming)

Compression ratio, e.g., 3.5MB Vs 50MB for mp3 song

Generality

Lossy:

Same as lossless but also need to judge how good the reconstructed message is

CODING

A message or file consists of *symbols*

A **code** assigns or maps **codewords** to source symbols

Common examples:

- Morse code (dots, dashes and spaces)
- ASCII code (1s and 0s, length 7)
- Barcodes (thin and thick vertical lines)
- Telephone codes

SAMPLE CODING: RUN LENGTH ENCODING (RLE)

- Uses the fact that in certain sorts of files we often get a sequence of repeated symbols
- Encodes a run of symbols as **{symbol, count}** by counting a “run” of the same character and storing the symbol along with the count
- Unlikely to occur in text files but common in many image and video files and binary files (sequences of 1s and 0s)



RLE EXAMPLES

aaaaaabbbbbbbbbbcccccc

Encoded as: {'a', 5}, {'b', 9}, {'c', 5}

11111111110000000011111111111111111

Encoded as ?

aabbbaaaaaaaaaaeeffffggggghiiii

Encoded as ?

For text, we normally work with fixed length encoding

Fixed length encoding uses same number of bits for each symbol

For example, ASCII:

Char	Decimal	Code
a	97	1100001
b	98	1100010
c	99	1100011
d	100	1100100
<i>etc</i>		

FOR EXAMPLE

Message M contains
characters/symbols:

110001011000011100100

Char	Decimal	Code
a	97	1100001
b	98	1100010
c	99	1100011
d	100	1100100
etc		

What is message?

How many bits in message?

TRY USING A SHORTER CODE (still fixed length)

char	code
a	000
b	001
c	010
d	011

And now same message again?

HOW TO KNOW THE LENGTH REQUIRED?

If N = number of different symbols

Then $\text{lower}(\log_2 N)$ length code required

e.g., In genomic sequences have only 4 codons: **a c t g**

$$\text{lower}(\log_2 4) = 2$$

So 2 bit code sufficient:

Say, **a** = 00, **c** = 01, **t** = 10, **g** = 11

EXAMPLE:

If $a = 00$, $c = 01$, $t = 10$, $g = 11$

in a fixed length 2-bit code, decode the following:

0001100001001100101100

Note:

The decoder (us in this case) need to know code

VARIABLE LENGTH CODING

Variable length coding uses different length codes for different symbols

Example 1: Given the following variable length code:

a = 1

b = 01

c = 101

d = 011

Is it possible to decode:

1011

PROBLEM?

Which is the correct decoding?

Solutions:

1. Add special stop/separator symbol
2. Choose code which can always be uniquely decoded by choosing codes where no code is a prefix of another

SAME EXAMPLE AGAIN (4 SYMBOLS) BUT WITH DIFFERENT CODE

$a = 1$	$b = 01$
$c = 000$	$d = 001$

Is any code a prefix of another?

Decode:

- 1001
- 1011
- 0001
- 0111

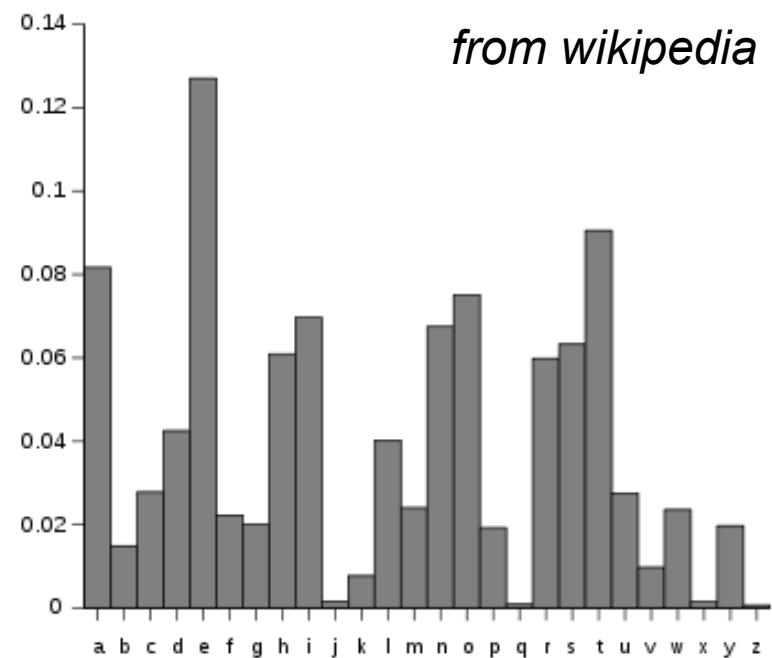
TEXT COMPRESSION

One approach to **text compression** uses *probability distribution of the symbols* in the message/file, i.e.,

- **More frequent symbols/words versus shorter symbols/words**

LETTER FREQUENCY IN ENGLISH?

- Mostly calculated based on general human written text
- May be differences if:
 - file was generated by computer (e.g. server logs)
 - file was created for a very specific context, e.g an essay on zebras or x-rays or qatar



LETTER FREQUENCY IN ENGLISH?

In order of most frequent:

e t a o i n s h r d l u c m f w y p v b g k j q x z

How frequent is *frequent*?

The top-12 letters comprise about 80% of the total usage:

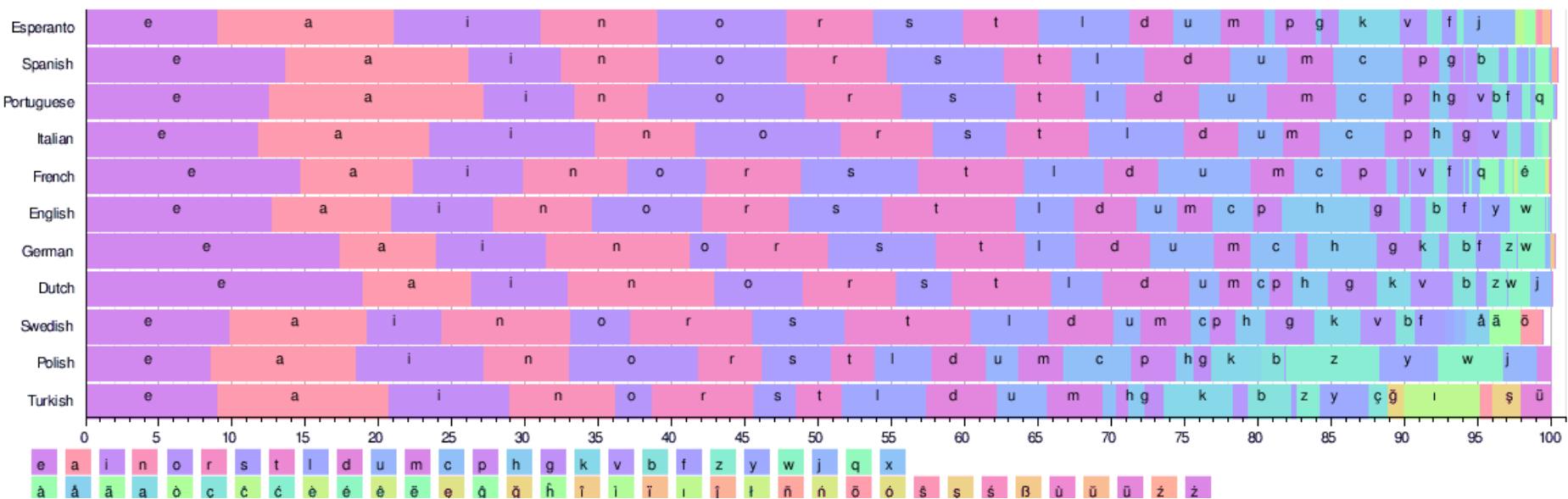
e t a o i n s h r d l u

The top-8 letters comprise about 65% of the total usage:

e t a o i n s h

LATIN LETTER FREQUENCY ACROSS SOME OTHER LANGUAGES

(http://en.wikipedia.org/wiki/Letter_frequency)



HUFFMAN CODING

- A **lossless** data compression technique
- Produces **optimal prefix codes** which are generated from a set of probabilities (based on frequency of occurrence) by the Huffman Coding Algorithm
- Guarantees **prefix property** – that is, no code is a prefix of any other code.
- Used as back-end of GZIP, JPEG, Brotli and many other utilities
- If good letter probabilities are available - and not **too costly** to obtain - then Huffman coding is a good compression technique and can achieve **an average of 2.23 bits per symbol**

Letter	English
a	8.17%
b	1.49%
c	2.78%
d	4.25%
e	12.70%
f	2.23%
g	2.02%
h	6.09%
i	6.97%
j	0.15%
k	0.77%
l	4.03%
m	2.41%
n	6.75%
o	7.51%
p	1.93%
q	0.10%
r	5.99%
s	6.33%
t	9.06%
u	2.76%
v	0.98%
w	2.36%
x	0.15%
y	1.97%
z	0.07%

OBTAINING LETTER PROBABILITIES?

1. Can use generic ones – derived for the language and domain:

e.g., English:

- e: 12.7
- t: 9.06
- a: 8.17
- o: 7.51
- i: 6.97

etc.

2. Can be the actual frequencies found in the text being compressed - this requires that a frequency table must be stored with the text (for decoding)

PREFIX CODE REPRESENTATION

The “trick” with Huffman coding is to represent the prefix codes using a **binary tree** where:

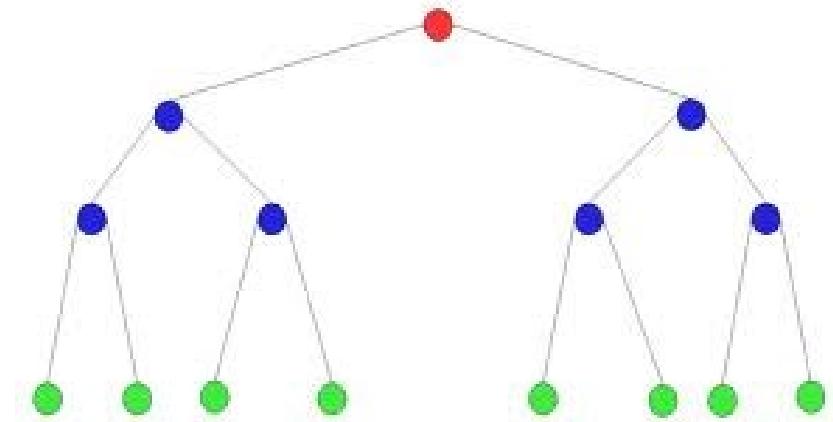
- each symbol is a leaf in the binary tree
- the code for each symbol is given by following a path from the root to the leaf and appending:
 - 0 for each left branch
 - 1 for each right branch

Note: by convention LHS is given 0 and RHS is given 1 - as long as encoder and decoder use the same labelling it does not matter

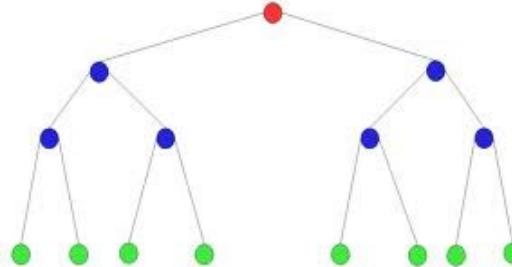
DEFINITION: BINARY TREE

A Binary tree consists of a set of **non-linear** nodes such that there is:

- One distinctive root node
- All other nodes are arranged such that each parent node can have at most 2 “child” nodes (a left and a right sub-node)



BINARY TREES

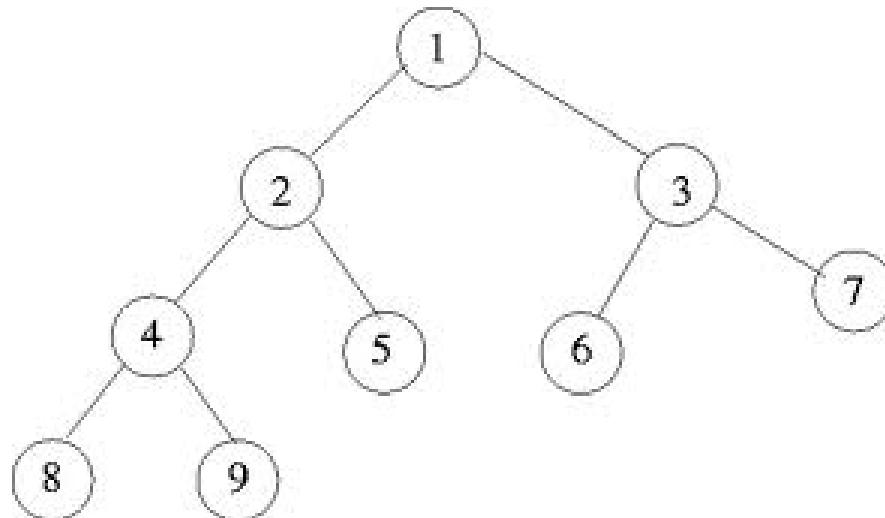


- The nodes with no child nodes (or sub-nodes) are often called ‘leaf nodes’
- The lines connecting the nodes are often called branches
- Paths are generally taken from the root node to the leaf nodes. At each stage can potentially choose to go left or right at a node and “follow” the branch to the next node.

EXAMPLES: List the paths ... list the nodes visited

From the root node (1) to:

- Node 8
- Node 6
- Node 5



CODE TREES (Weighted binary trees)

- Each leaf node represents a symbol
 - Each branch has a “weight” associated with it (either 0 or 1)
 - Left branches are weighted 0
 - Right branches are weighted 1
 - To find the code associated with the symbol:
 - Start at root node and keep appending the weights from root to the symbol as you follow path from root to leaf node of interest
- * This also gives the length of the code (the number of branches traversed).

EXAMPLE 1:

The code for symbol E is:

1111 (right,right,right,right)

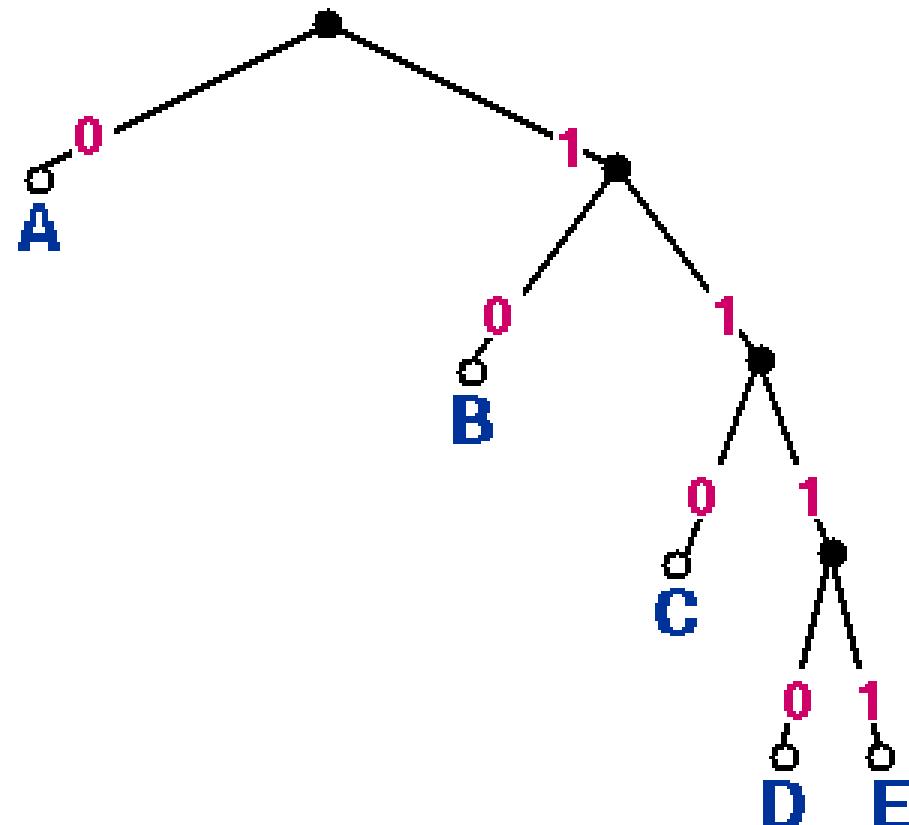
List the codes for each of
the other symbols:

A

B

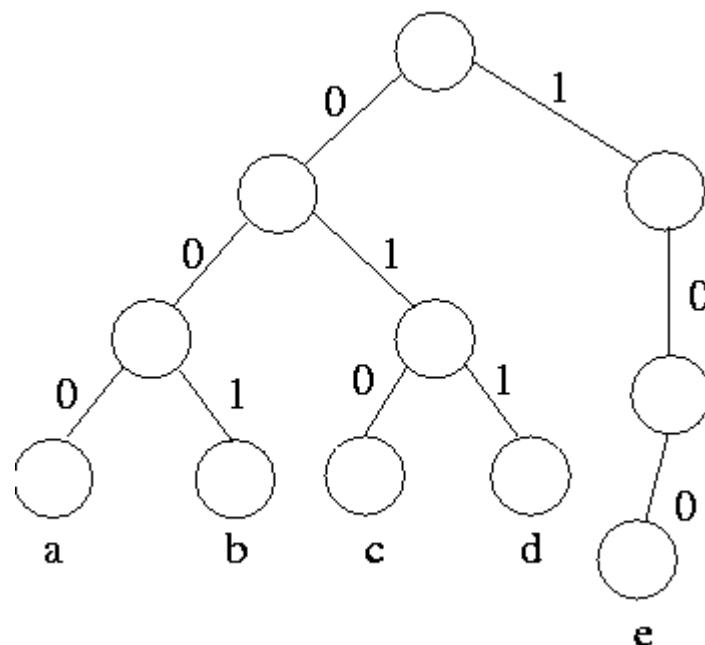
C

D



EXAMPLE 2:

What **length** (number of branches) are the codes for the symbols in the given code tree?



HUFFMAN COMPRESSION ALGORITHM

Input: Symbols (letters) and their probabilities (frequencies)

1. Create a trivial tree (node) for each letter
2. Assign a weight to each node – initially, the weight of each node is the frequency of the letter associated with that node
3. Sort (order) trees by weight (a priority queue), **smallest to largest**.
4. Decide on rule for ties: - will not affect code length but must be consistent with encoding/decoding stages. Our approach:
 - If there is a tie with single node trees, order (alphabetically) by letter (symbol)
 - If there is a tie otherwise, order by tree size (number of nodes in tree - smallest to largest)

5. while (more than one tree left in priority queue)

{

merge the two trees at the start of the priority queue (those with smallest weights) to create a new tree such that:

- Root of tree has, as its weight, the summation of the weights of the sub-trees
- the tree at the top of the queue is a **left sub-child** of root; the next tree is a right **sub-child** of root

place new tree back in queue in correct place (in sorted order)

}

6. Label edges of final tree (left 0; right 1)

Output: Huffman code tree from which can read codes for each letter

EXAMPLES

3. Given the following letters and their frequency, construct a Huffman code tree:

t a e h

10 5 15 3

4. Given the following message, find codes, using Huffman compression, for each unique letter in the message. Calculate frequencies of each letter from message:

this is mississippi

EXAMPLE 4: this is mississippi

Frequencies:

h	m	t	p	i	s
1	1	1	2	6	6

HUFFMAN DECOMPRESSION ALGORITHM

Input: letters and their frequencies and sequence of binary codes

Approach:

1. Build Huffman tree using **exact** same algorithm as was used for compression
2. For each encoded symbol, follow path from root node to leaf node, based on current number (1 or 0), until you reach symbol at leaf node.

Output: original message

EXAMPLE 5: Decompress the messages

Given the probabilities of the following 5 symbols:

- $P(a) = 0.12$
- $P(b) = 0.4$
- $P(c) = 0.15$
- $P(d) = 0.08$
- $P(e) = 0.25$

What are the words represented by the following Huffman codes?

01010

0101111110

Huffman codes of 27 english symbols (includes -)

Copyright Cambridge University Press 2003. On-screen viewing permitted. Printing not permitted. <http://www.cambridge.org/0521642981>
You can buy this book for 30 pounds or \$50. See <http://www.inference.phy.cam.ac.uk/mackay/itila/> for links.

100

5 — Symbol Codes

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
a	0.0575	4.1	4	0000
b	0.0128	6.3	6	001000
c	0.0263	5.2	5	00101
d	0.0285	5.1	5	10000
e	0.0913	3.5	4	1100
f	0.0173	5.9	6	111000
g	0.0133	6.2	6	001001
h	0.0313	5.0	5	10001
i	0.0599	4.1	4	1001
j	0.0006	10.7	10	1101000000
k	0.0084	6.9	7	1010000
l	0.0335	4.9	5	11101
m	0.0235	5.4	6	110101
n	0.0596	4.1	4	0001
o	0.0689	3.9	4	1011
p	0.0192	5.7	6	111001
q	0.0008	10.3	9	110100001
r	0.0508	4.3	5	11011
s	0.0567	4.1	4	0011
t	0.0706	3.8	4	1111
u	0.0334	4.9	5	10101
v	0.0069	7.2	8	11010001
w	0.0119	6.4	7	1101001
x	0.0073	7.1	7	1010001
y	0.0164	5.9	6	101001
z	0.0007	10.4	10	1101000001
-	0.1928	2.4	2	01

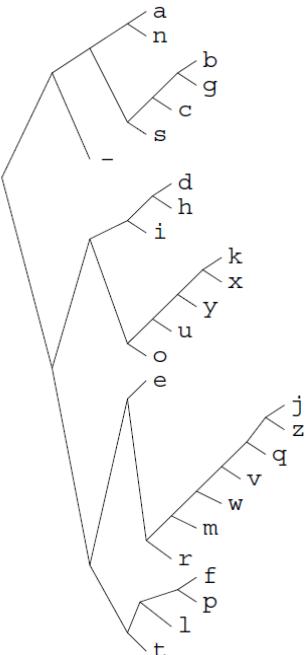


Figure 5.6. Huffman code for the English language ensemble (monogram statistics).

ARITHMETIC CODING

Arithmetic coding encodes a stream of symbols (rather than a single symbol) as a single floating point number, in the range from 0.0 to 1.0

Also an example of *lossless* data compression

Stream text encoding is more common now

ARITHMETIC CODING APPROACH

Input: message and symbols and their frequencies

General Approach: Work with intervals and sub-intervals where each interval represents a proportion relative to the probability of the occurrence of the message

Output: real number in range [0, 1.0)

ASIDE: RANGES ...

[0, 1] : 1 and 0 included in range

[0, 0.5): 0 included in range; 0.5 not

ARITHMETIC ENCODING ALGORITHM

1. Begin with interval = [0.0, 1.0)
2. Get all symbols and their probabilities of occurrence
3. Order the symbols – smaller frequency first, alphabetically if of the same frequency
4. Place symbols from message in a queue, in the frequency order given from step 3

5. while symbols left in queue{

- For current interval, divide the interval according to the probabilities of all symbols occurring and the order of these symbols (step 3), starting at lowest range of interval (e.g., 0.0)
- Let current message symbol = symbol at top of queue
- Find in which interval current message symbol lies, this becomes new interval and divide interval as before
- Get next message symbol

}

EXAMPLE 6:

Inputs:

- symbols are: a c r
- Probabilities are: $P(r) = 0.2$, $P(a) = 0.4$, $P(c) = 0.4$
- Message = car

Approach:

Order symbols according to probabilities first, alphabetically second:

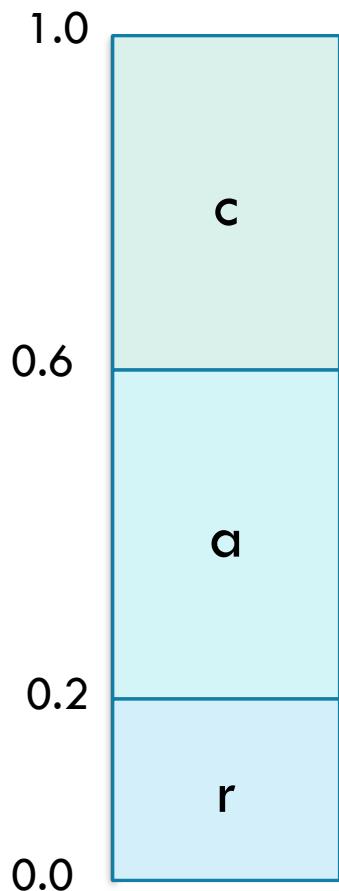
- r a c

First iteration of loop:

- Assign each symbol to the an interval in range:
 - r -> [0.0, 0.2)
 - a-> [0.2, 0.6)
 - c->[0.6, 1.0)
- First symbol of message is c, so new interval is [0.6, 1.0)

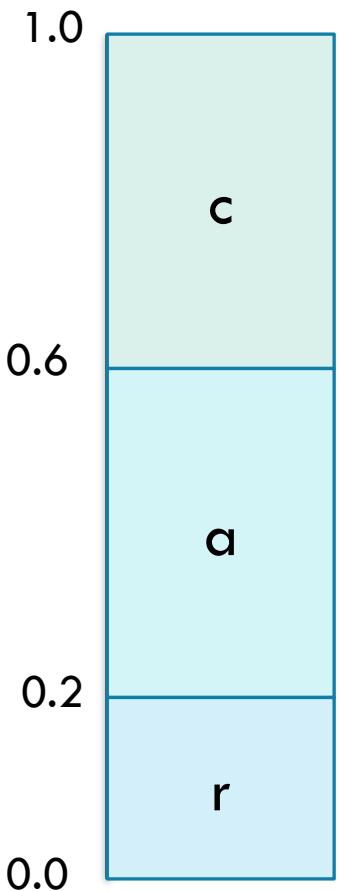
MESSAGE: car

1st interval: 0.0 to 1.0



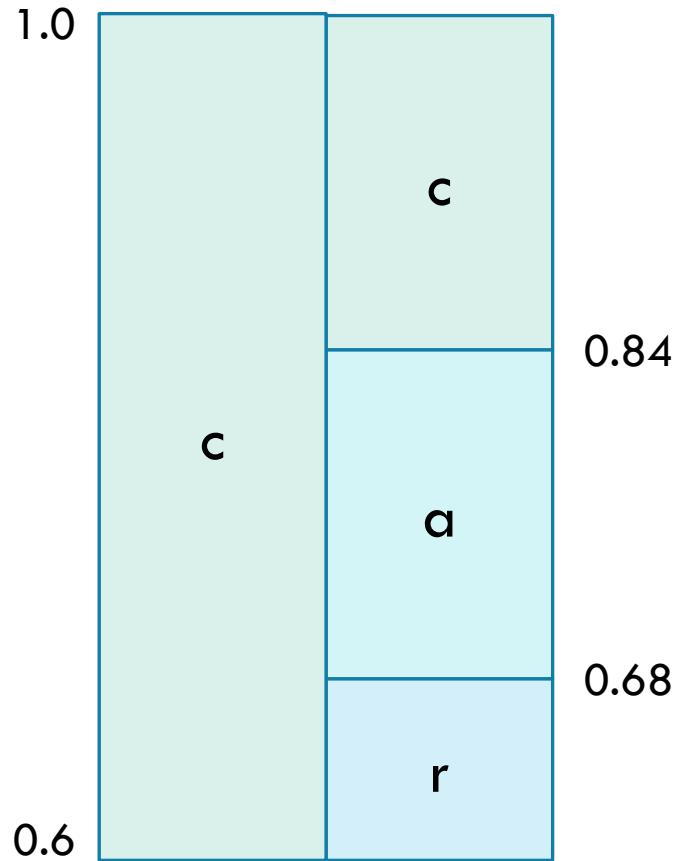
1st symbol = “c”

Interval with “c”: [0.6-1.0)



2nd symbol = “a”

Interval with “ca”: [0.68,0.84)



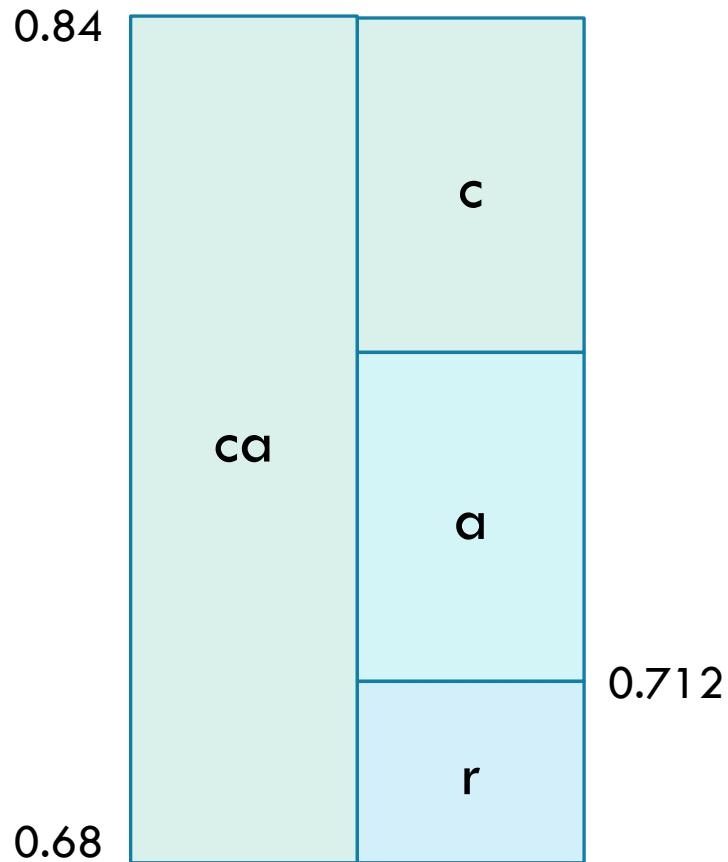
$$r: .2 \cdot .4 = 0.08$$

So cr interval is represented by: $[0.6, 0.6+0.08) = [0.6, 0.68)$

$$\alpha: .4 \cdot .4 = 0.16$$

So ca interval is represented by:
 $[0.68, 0.68+0.16) = [0.68, 0.84)$

3nd symbol = “r”
Interval with “car” : [0.68,0.712)



r: $.2 * .16 = 0.032$
So car interval is
represented by: $[0.68 + .68 + 0.032) =$
[0.68, 0.712)

PRACTICAL CONCERNS ...

The longer the text stream (sequences/message) encode the more precise the interval to encode it becomes but machines have finite precision.

The solution is to limit the length of strings encoded at any one time ... so that strings of a certain length are encoded where the maximum length is determined by the precision available.

EXAMPLE 7:

Input:

- symbols are: A, B, C
- probabilities are $P(A) = 0.5$, $P(B) = 0.25$, $P(C) = 0.25$
- Message is CAB

First iteration of loop:

- Assign each symbol to the an interval in range:
 - B -> [0.0, 0.25)
 - C-> [0.25, 0.5)
 - A->[0.5, 1.0)
- First message symbol is C, so new interval is ???
- etc.

INTERVAL SENT?

In reality rather than transmit the interval (2 numbers), a real-valued number (or rather binary representation of it) is transmitted instead

With this approach, for decoding, the length of the string is needed also (fixed length can be used so that it only has to be transmitted once)

e.g., for interval **[0.68, 0.712)**

Some number around 0.71 can be transmitted, also knowing string is of length 3

What is 0.71... in binary?

ALGORITHM: DECODING

Input: binary number and symbol length
(assuming symbols, their order and frequencies known)

General Approach: Get real number, follow the same approach as for encoding but at each stage consider next current digit to find correct interval

Stop when at sub-interval of the correct length

Output: symbols from message

EXAMPLE 8:

Inputs:

- Symbols are **a**, **c**, **r**
- Symbols are ordered as: **r a c** based on the probabilities of: $P(r) = 0.2$, $P(a) = 0.4$, $P(c) = 0.4$
- Compressed message is represented by binary **0.01** and all messages are of length 3 (3 symbols)
- What is decompressed message?

SUMMARY: IMPORTANT TO KNOW

Compression: encoding and decoding

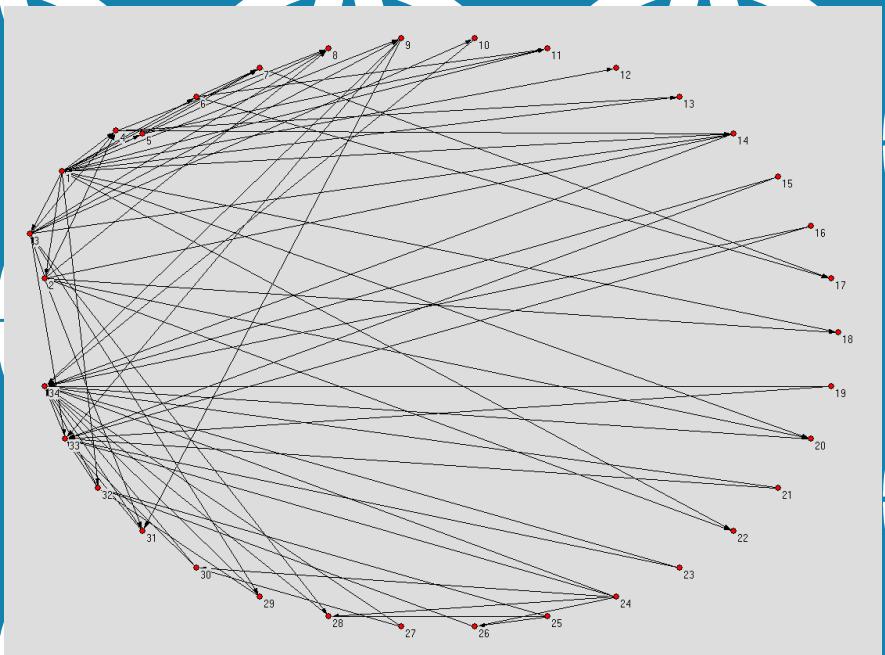
(Run Length Encoding)

Variable length and fixed length codes

Prefix Codes

Two text approaches based on letter frequencies:

- Huffman Coding
- Arithmetic Coding
- Worked examples for encoding and decoding using both techniques

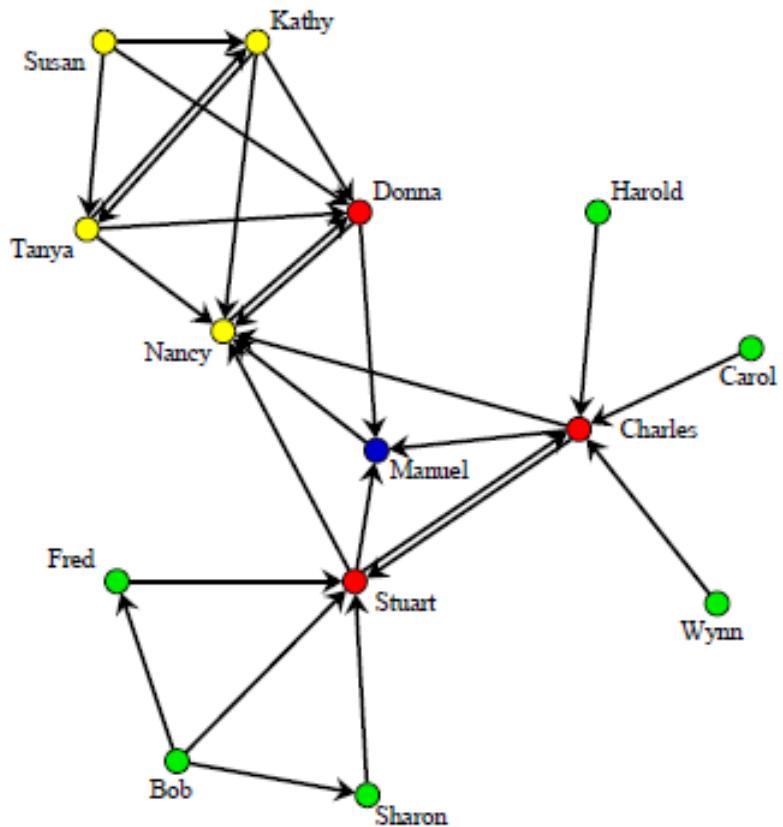


SOCIAL NETWORK ANALYSIS

CT102:
Information
Systems

DEFINITION: SOCIAL NETWORK

Views social relationships **between people** in terms of a graph or network



A SOCIAL NETWORK HAS ...

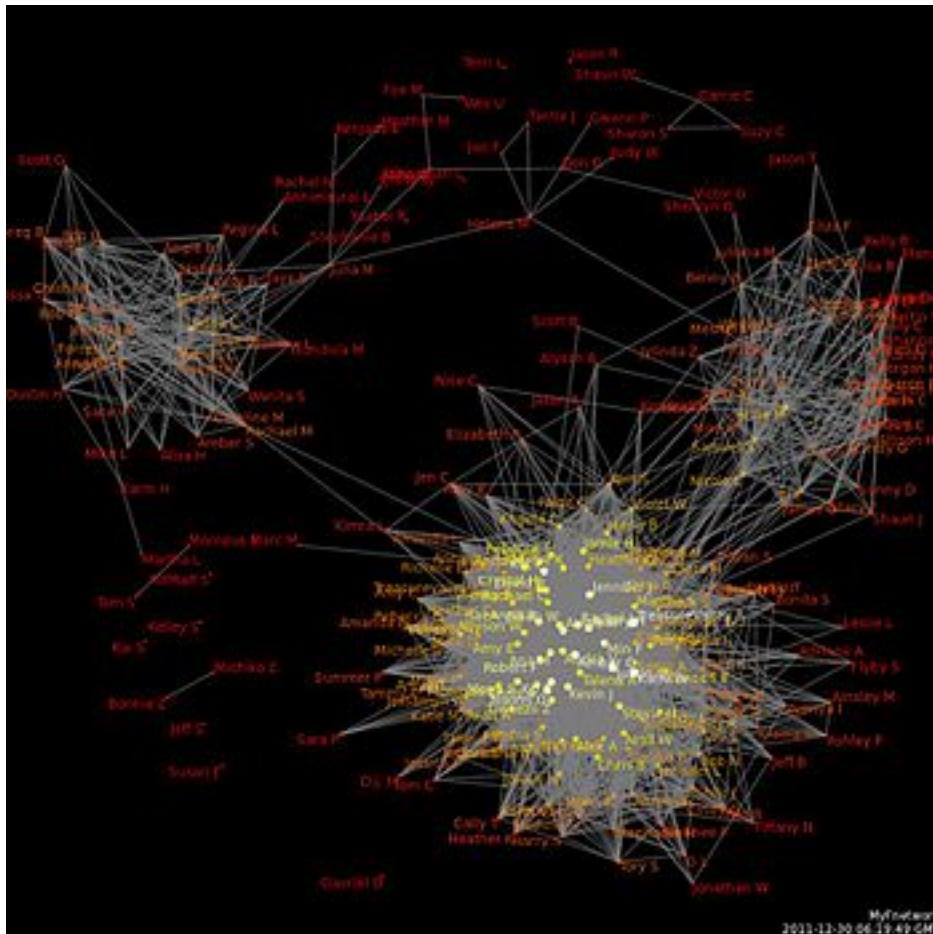
Nodes represent people/actors/entities. Often represented as points:

- Data is associated with nodes
- May be **one mode** (nodes are of the same type (e.g. people) or **two mode** (nodes represent two different things - e.g. people and items)

Edges or *ties* connect two nodes and represent the social relationships between the nodes (by a line):

- May be different types of edges
- May have weight
- Can be directed (with arrow) or undirected (when the connection means the same to both actors)

EXAMPLE



A social network diagram displaying friendship ties among a set of Facebook users.

(wikipedia)

SOCIAL NETWORK DATA REPRESENTATION

Two main approaches:

1. Adjacency Matrix (sociomatrix)
2. Edge List

ADJACENCY MATRIX

Equal number of rows and columns

Number of rows and columns equal the number of actors

Elements in the matrix represent the relationship between the actors (binary or real number)

Conventionally, the relationship is read from the row (i) actor to the column (j) actor and (i,j) represents the presence/absence of an edge and – if used – the edge weight

Main diagonal usually blank (self-relationships) or contains 0

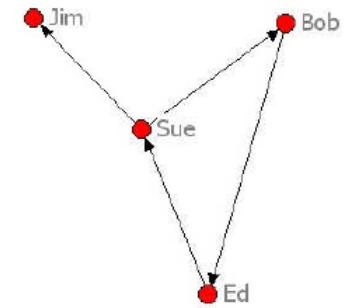
NOTE: DIFFERENCE BETWEEN ATTRIBUTE AND SOCIAL (RELATIONSHIP) DATA

With database systems, each tuple holds data for that instance or individual

Name	Dept	Gender	Salary
Ed	14	M	50000
Sue	15	F	70000
Jim	16	M	65000
Bob	17	M	15000

Matrices/Grids are used to represent graphs to show *relationships* between individuals rather than attributes of individuals

	Ed	Sue	Jim	Bob
Ed	-	1	0	0
Sue	0	-	1	1
Jim	0	0	-	0
Bob	1	0	0	-



EDGE LISTS

If binary connections exist:

- Usual to store connections between nodes in pairs
(id1, id2)
- Can also store as a list of the links for each id (id1, id2, id3, id4)
- If undirected then do not store relationship in both directions
- If directed, the convention is generally that the edge is **from** the first entry **to** the second entry, e.g., from id1 to id2

If weighted connections:

- usual to store a triple, (id1, id2, weight)

May need to supplement with a list of nodes if some do not have any edges

EXAMPLE

Draw the (undirected) graph and adjacency matrix represented by the following edge list:

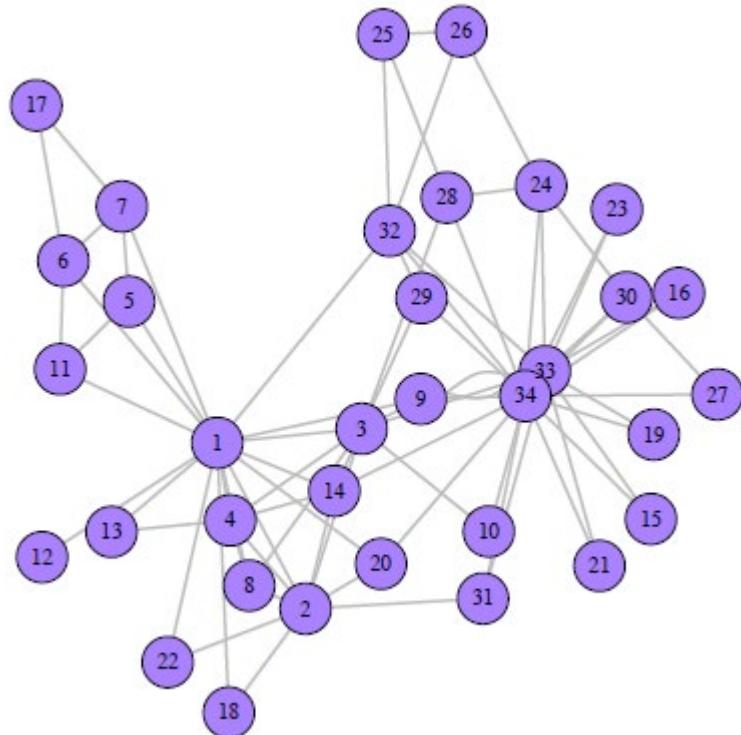
(a, b)

(a, c)

(b, c)

(b, d)

EXAMPLE: SOCIAL NETWORK OF FRIENDSHIP LINKS WITH 34 NODES



Taken from:

Zachary, Wayne W.: An Information Flow Model for Conflict and Fission in Small Groups, Jnl pf Anthropological Research, 33(4), 452-473, 1977

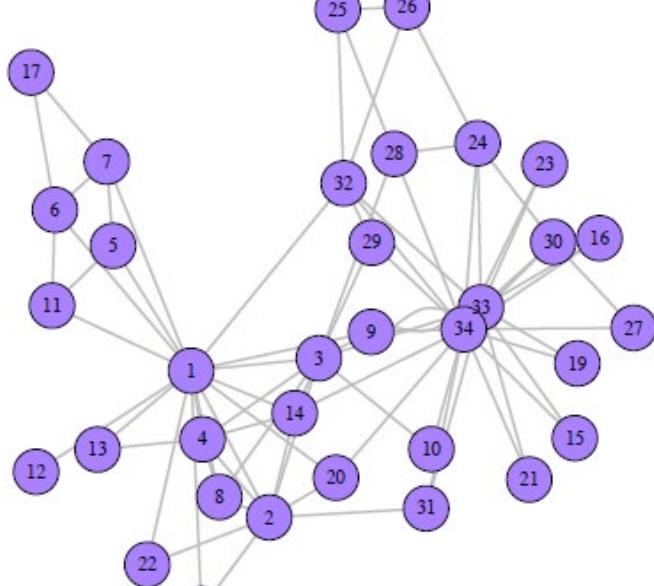
Network file available on Blackboard:

(format edge list)

`zachary.net`

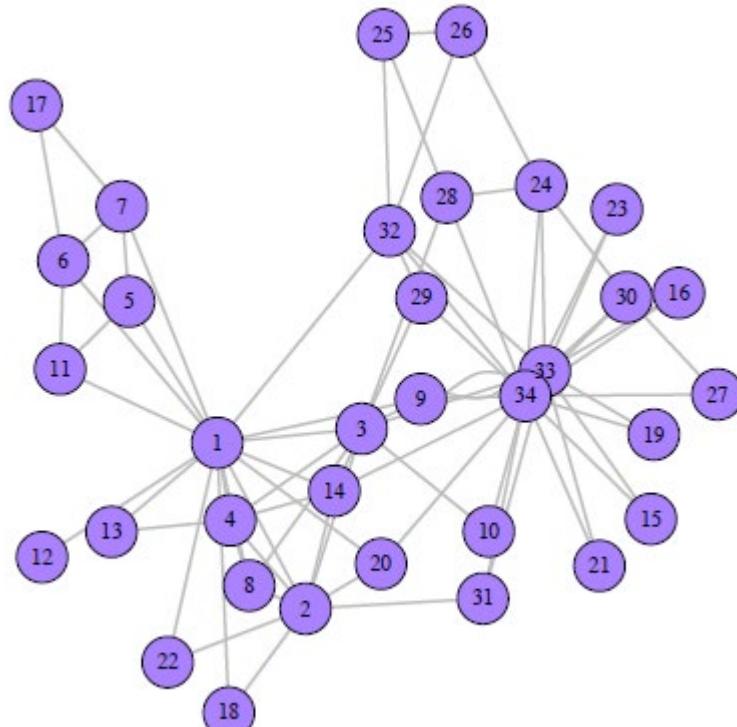
MATRIX REPRESENTATION OF SAME EXAMPLE

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	
1	0	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0
4	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
17	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
22	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
31	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
32	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
33	0	0	1	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0



EDGE LIST REPRESENTATION OF SAME EXAMPLE (ZACHARY.NET)

1	32	3	14	19	34
1	22	3	9	19	33
1	20	3	10	20	34
1	18	3	33	21	34
1	18	3	29	21	33
1	14	3	28	23	34
1	13	3	8	23	33
1	12	3	4	24	30
1	11	4	14	24	34
1	9	4	13	24	33
1	8	4	8	24	28
1	7	5	11	24	26
1	6	5	7	25	32
1	5	6	17	25	28
1	4	6	11	25	26
1	4	6	7	26	32
1	3	7	17	27	34
1	2	9	34	27	30
2	31	9	33	28	34
2	22	9	31	29	34
2	20	10	34	29	32
2	18	14	34	30	34
2	14	15	34	30	33
2	8	15	33	31	34
2	4	16	34	31	33
2	3	16	33	32	34
				32	33
				33	34

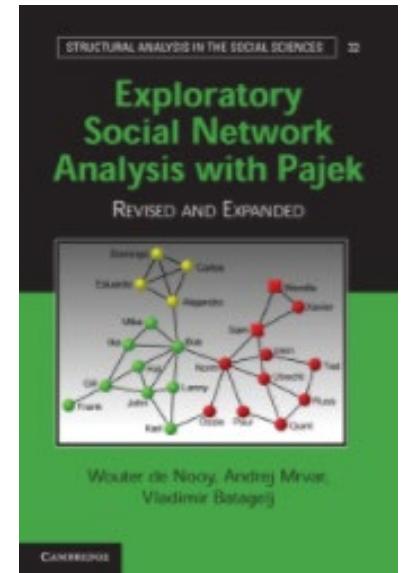


ASIDE:

Note that any type of graph/network can be represented by an edge list and matrix

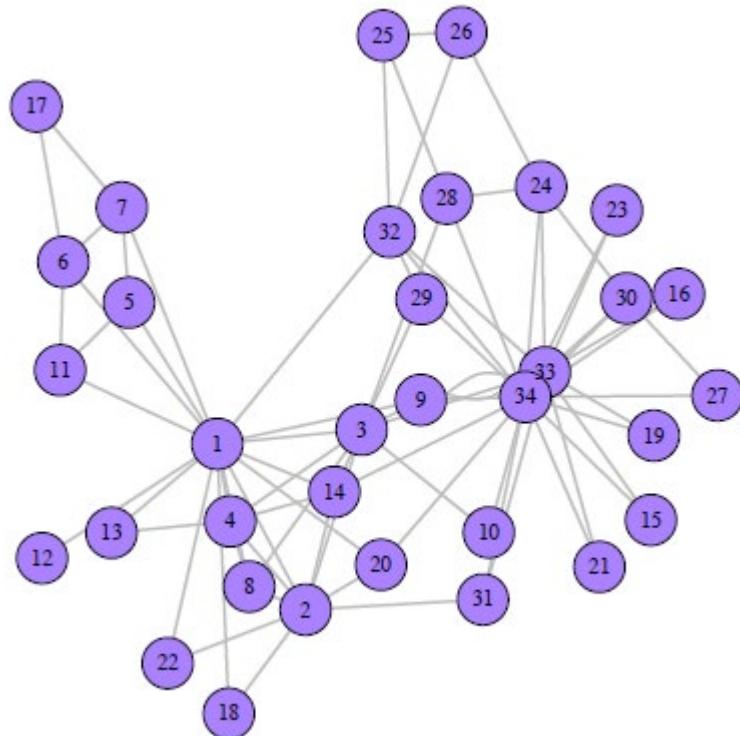
SOFTWARE FOR OPERATIONS ON NETWORKS

- For large social networks, many software tools exist to perform network analysis operations on networks, negating the need for us to program these ourselves. However as always Important to know what the operations are and how they work!



WHAT IS OF INTEREST AT A SNA LEVEL?

- Properties of Nodes
- Properties of Relationships
- Properties of Network



1. (SOME) PROPERTIES OF NODES

- 1.1. Degree centrality
- 1.2. Betweenness centrality
- 1.3. Closeness centrality
- 1.4. Eigenvector centrality

1.1. DEGREE CENTRALITY:

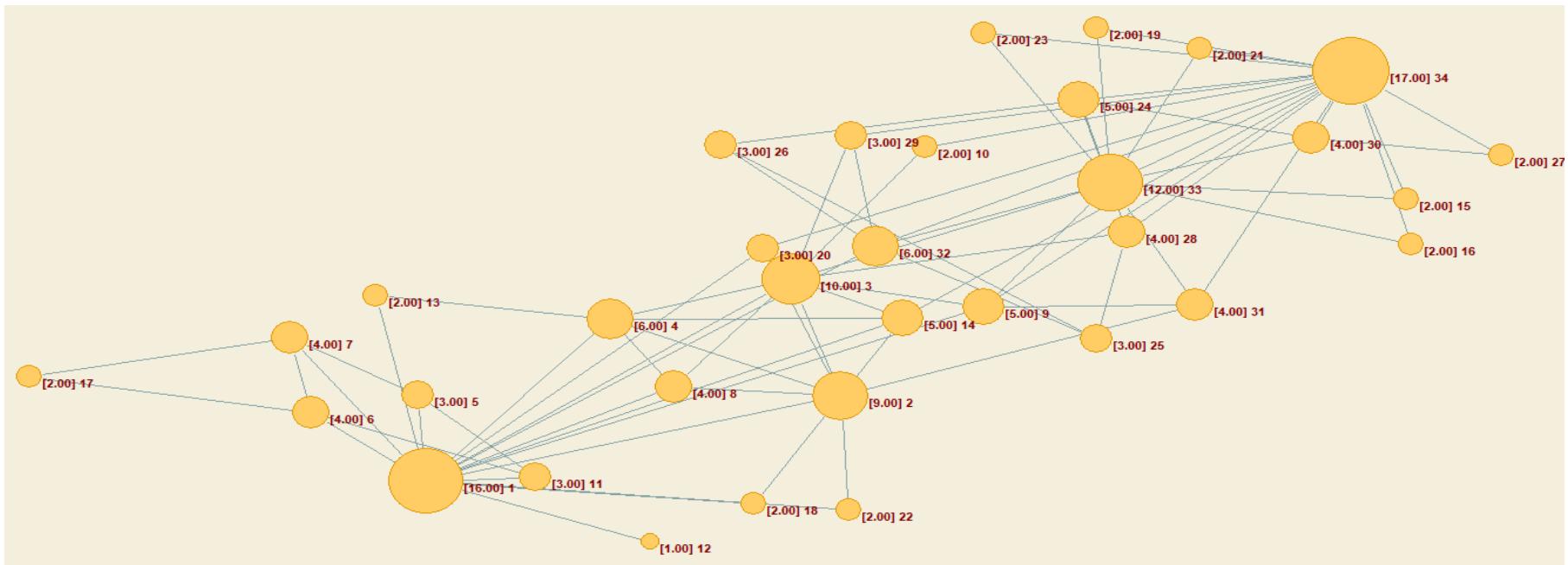
Degree: number of edges a node has

If directed network then can distinguish between:

- Indegree: number of **in** edges to a node from other nodes
- Outdegree: number of **out** edges from a node to other nodes

Often these measures are averaged over all nodes in the graph

DEGREE CENTRALITY OF ZACHARY EXAMPLE NETWORK



NOTE 1:

Degrees can be calculated by a row and/or column summation using the adjacency matrix representation

NOTE 2:

Given edge list representation how would you calculate degree of each node using a C program?

1 32	3 14	19 34
1 22	3 9	19 33
1 20	3 10	20 34
1 18	3 33	21 34
1 14	3 29	21 33
1 14	3 28	23 34
1 13	3 8	23 33
1 12	3 4	24 30
1 11	4 14	24 34
1 9	4 13	24 33
1 8	4 8	24 28
1 7	5 11	24 26
1 6	5 7	25 32
1 6	6 17	25 28
1 5	6 11	25 26
1 4	6 7	26 32
1 3	7 17	27 34
1 2	9 34	27 30
2 31	9 33	28 34
2 22	9 31	29 34
2 20	10 34	29 32
2 18	14 34	30 34
2 18	15 34	30 33
2 14	15 33	31 34
2 8	16 34	31 33
2 4	16 33	32 34
2 3		32 33
		33 34

AVERAGE IN/OUT DEGREE

The average degree can also be used to give a comparison across a social network

Average is calculate by summing degrees (*in* or *out* or *both*) of each of the N nodes and dividing by the number of nodes N

The ratio of *in* and *out* Degrees to averages are often used to give a measure of an individual's influence

PATHS AND REACHABILITY

An individual (node), A, is “reachable” by another, B, if there exists a set of edges (connections) by which we can move (traverse) from B to A.

If A is reachable by B we say that a *path* exists between B and A.

The *number of edges* needed to be traversed in order to move from B to A is called the *path length*

* If considering a directed graph then it may be possible that B is reachable by A but A is not reachable by B. *

DISTANCE

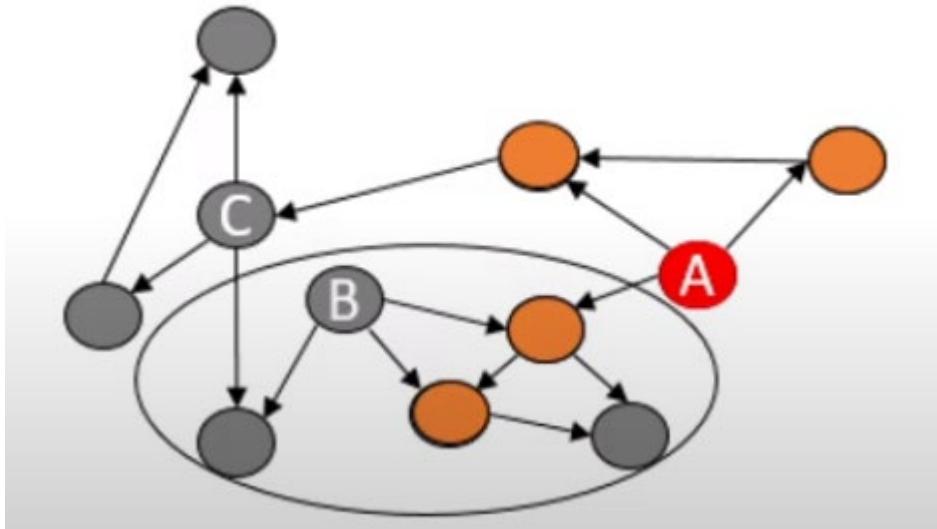
Distance concentrates on the length of paths (number of edges traversed) from nodes (individuals) in the network

Often interested in:

- The average distance between nodes
- The shortest path between nodes (geodesic distance)
- The **diameter** of a network - the longest of all shortest paths between all nodes (shortest distance between the two most distant nodes)

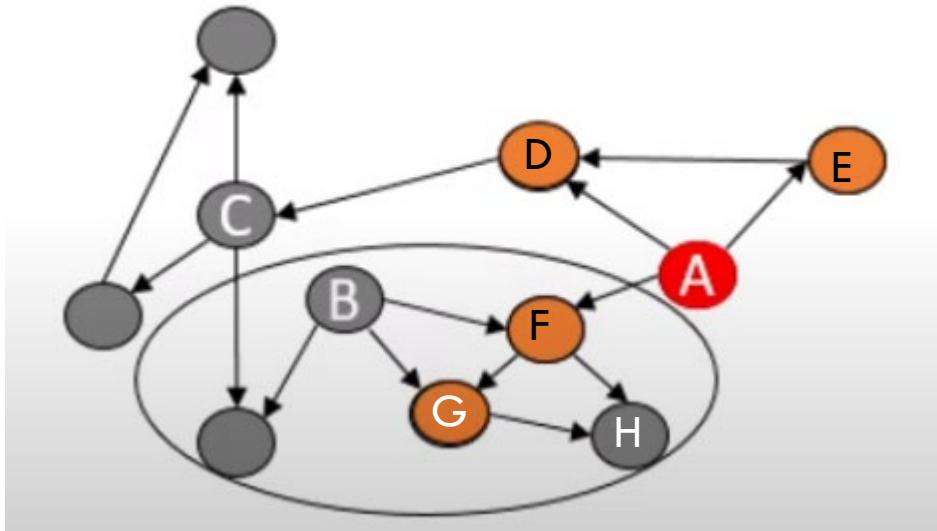
* Distance is used as a measure of individual influence as well as an approach to recommendation *

EXAMPLE: FINDING INFLUENTIAL NODES



Here node A, B and C have the highest outdegrees
But node A is the only node that has the most paths to all other nodes in the network (all except B)
So A is most influential

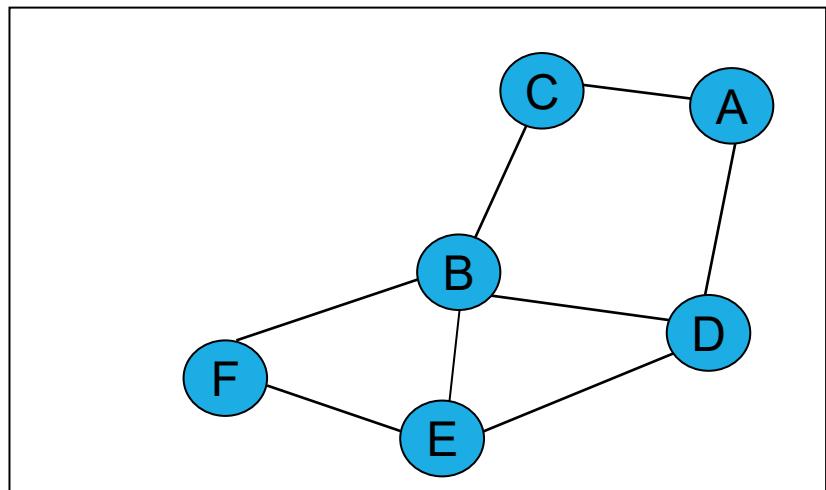
EXAMPLE: Making Recommendations to A



Consider the nodes that A is connected to as “friends”: D, E, F
Recommend the nodes that the friends are connected to (friends-of-friends): C, G and H

EXAMPLE

Given A as the starting node and F as the final node what is the (length of) the shortest path from A to F? (with no revisiting of nodes)



EXAMPLE: DIAMETER OF ZACHARY EXAMPLE NETWORK

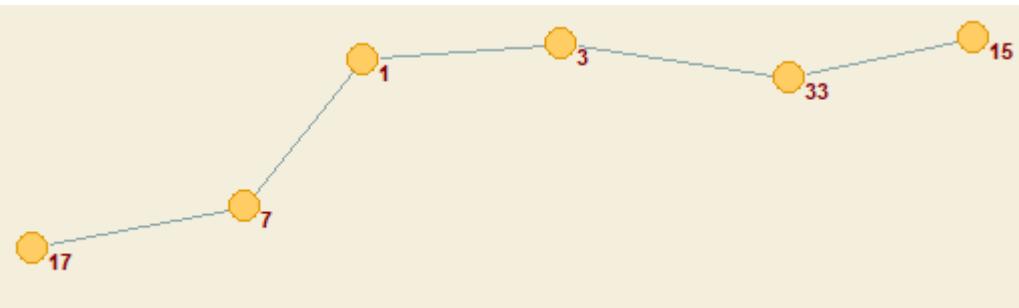
```
Searching the longest shortest path in 1. C:\Users\josep\Lectures\CT102- Infor
```

```
Working...
```

```
Result:
```

```
The longest shortest path from 15 (15) to 17 (17). Diameter is 5.
```

```
Time spent: 0:00:00
```



FINDING PATHS

Path and path lengths can be found by matrix multiplication

If A is an adjacency matrix then A^2 will find all paths of length 2

e.g.,

If $A =$

0 1 0 1

1 0 1 1

0 1 0 0

1 1 0 0

PATHS OF LENGTH 2 ...

$$\begin{array}{c|c|c|c} 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 \end{array} \times \begin{array}{c|c|c|c} 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 \end{array} =$$

$$\begin{array}{c|c|c|c} 2 & 1 & 1 & 1 \\ \hline 1 & 3 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 2 \end{array}$$

1.2. BETWEENNESS CENTRALITY

Quantifies the number of times a node acts as a **bridge** along the shortest path between two other nodes

Nodes that have a high probability of being on a randomly chosen shortest path between two randomly chosen nodes have a high betweenness centrality

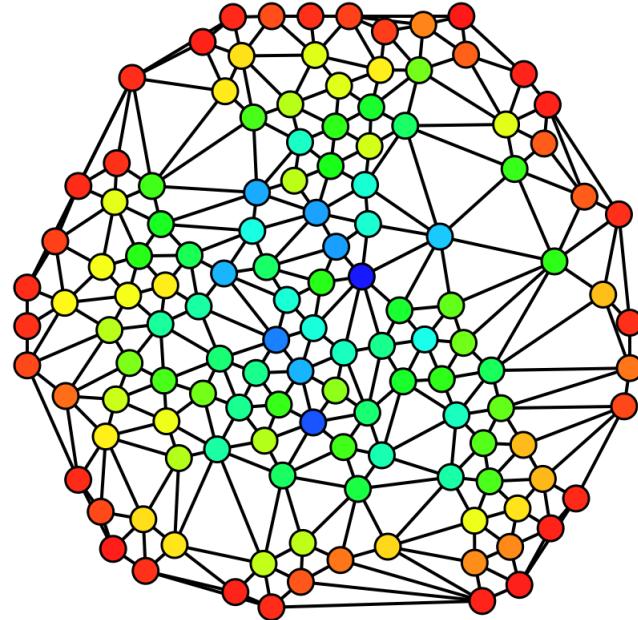


Image By Claudio Rocchini

<https://commons.wikimedia.org/w/index.php?curid=1988980>

Hue shows the node betweenness
(from red = 0 to blue = max)

CALCULATING BETWEENNESS

1. For each pair of nodes (s,t) , compute the shortest paths between them
2. For each pair of nodes (s,t) , determine the fraction of shortest paths that pass through the vertex in question
3. Sum this fraction over all pairs of vertices (s,t)

A large value usually indicates that a node is structurally central and a low value indicates that a node is structurally peripheral

EXAMPLE

Shortest Paths

- A, B = 1
- A, C = 2
- A, D = 2
- B, D = 1
- B, C = 1
- C, D = 1

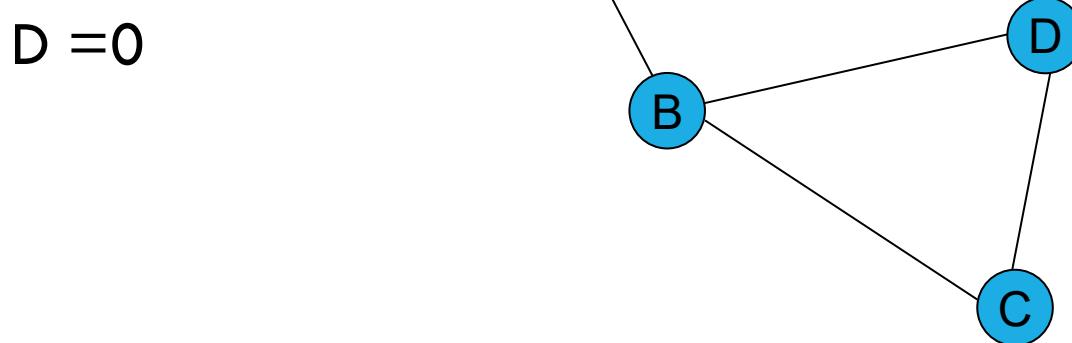
Paths Through Vertex

$$A=0$$

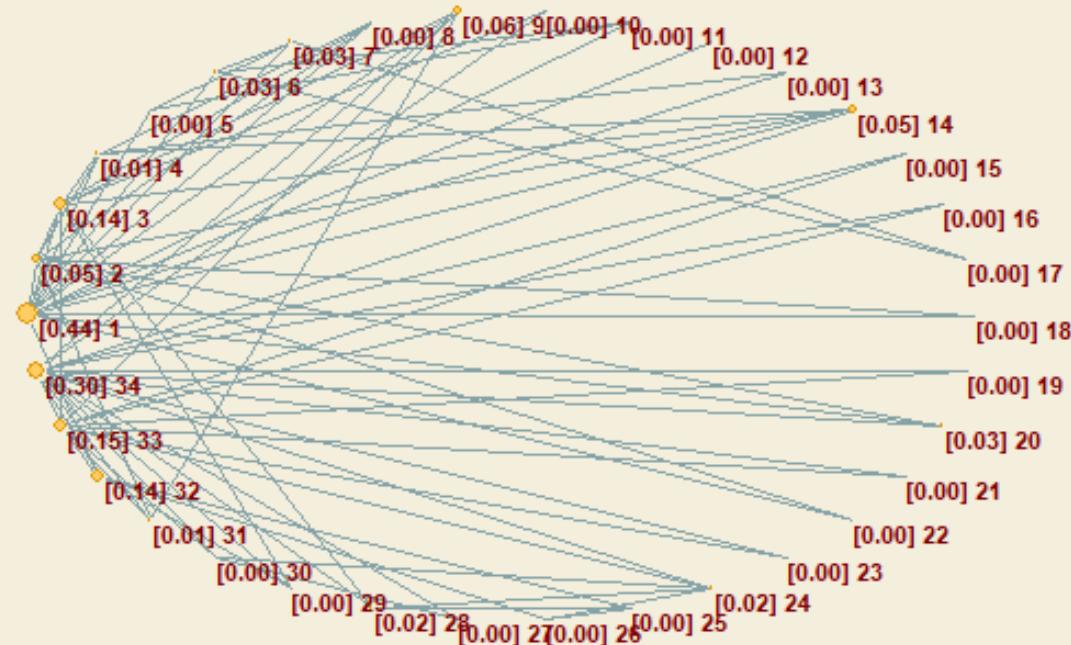
$$B=2/6 = 0.33$$

$$C=0$$

$$D=0$$



Betweenness centrality of zachary example network



Nodes with highest betweenness centrality:

- Node 1 (0.44)
- Node 34 (0.44)
- Node 33 (0.15)

Many nodes have betweenness centrality of 0

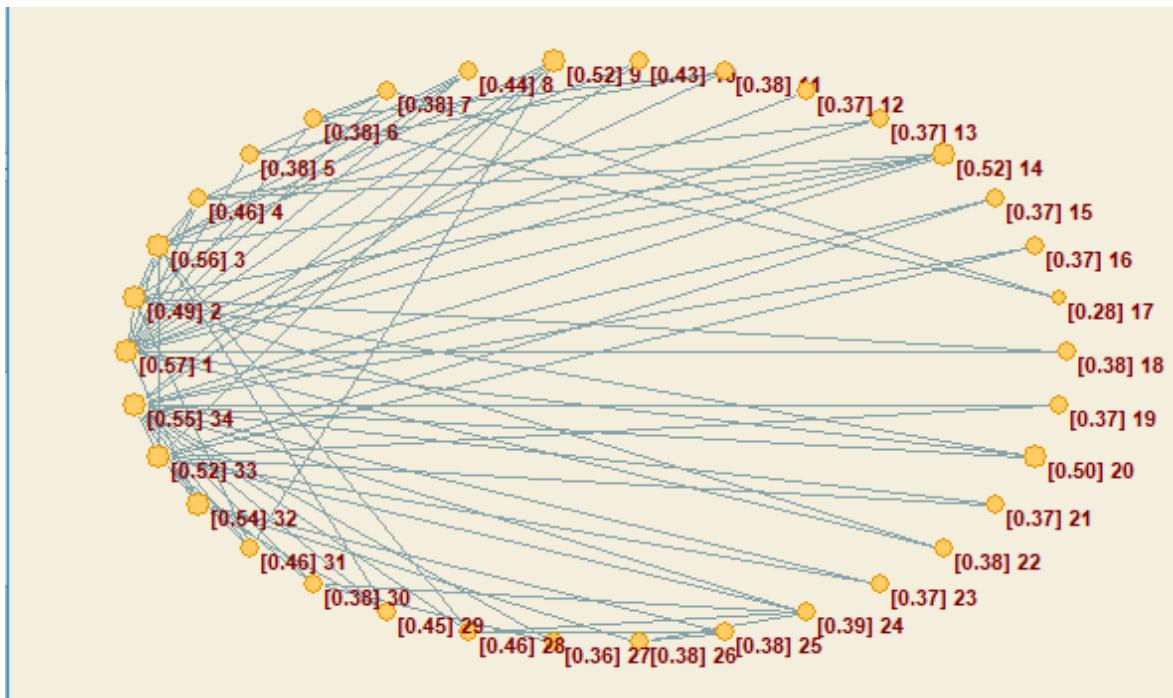
1.3. CLOSENESS CENTRALITY

Measures the path length between a particular node and all other nodes in the network

For any node:

$1 / (\text{sum of all shortest paths from a node to all others})$

Closeness centrality of Zachary example network



Nodes with highest closeness centrality:

- Node 1 (0.57)
- Node 34 (0.55)
- Node 3 (0.56)
- Node 32 (0.54)
- Node 33 (0.52)
- Node 8 (0.52)
- Node 20 (0.5)

Note no node has 0 for closeness centrality

1.4. EIGENVECTOR CENTRALITY

A variant of the [Page Rank](#) Algorithm where a node is considered important if it is linked to by other important nodes

Calculated using adjacency matrix representation

If A is matrix then looking for eigenvalue (λ) such that :

$$A v = \lambda v$$

where A is adjacency matrix

- v is eigenvector
- λ is eigenvalue (usually pick largest)

2. PROPERTIES OF RELATIONSHIPS

- 2.1. Reciprocity:** the tendency that if an edge (i,j) exists, then an edge (j,i) will also exist (for directed networks only)
- 2.2. Transitivity:** the tendency of friends-of-friends to be friends
- 2.3. Preferential Attachment (popularity):** the tendency for nodes that are already central to gain more connections at a greater rate than nodes which are less central
- 2.4. Clique:** a densely connected group – a subset of nodes in a network such that every two nodes in the subset are connected

3. PROPERTIES OF NETWORK

3.1. Edge density

3.2. Clustering Coefficient

3.3. Homophily: the tendency of actors who are similar on a particular trait to form ties (“birds of a feather ... ”)

3.4. Modularity: existence, or not, of community structure – clusters that are not densely connected to others but densely connected within a cluster

3.5. Diameter: the longest of all calculated shortest path between actors

3.6. Network Type: classification of different types of networks

3.1. EDGE DENSITY



1220 nodes

19139edges

Fully connected social networks (everyone connected to everyone) are rare in most cases

However, it is useful to see what percentage of all possible edges do exist ... this is called the density of edges

Edge Density definition:

- Ratio of the number of actual edges to the number of possible edges which exist in a network

NUMBER OF POSSIBLE EDGES

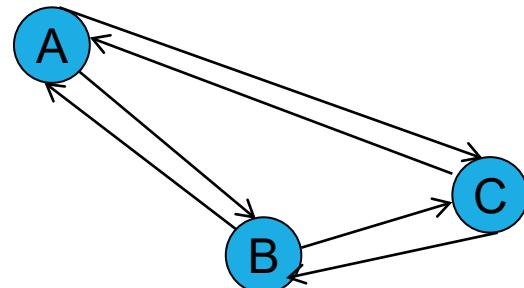
For a **directed** social network where number of nodes = N, the number of possible edges is:

$$N*(N-1)$$

(no self ties)

e.g., if N = 3:

$$\#\text{possible edges} = 3*2 = 6$$



NUMBER OF POSSIBLE EDGES

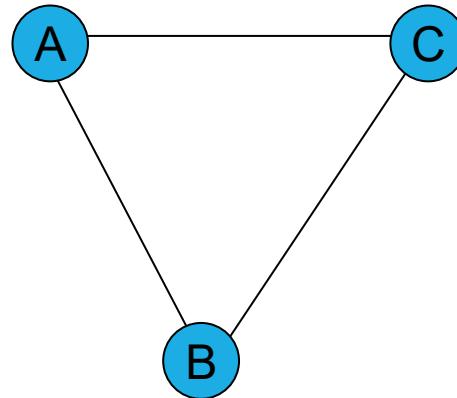
For an **undirected social network** with number of nodes = N , the number of possible edges is then half this:

$$N*(N-1)/2$$

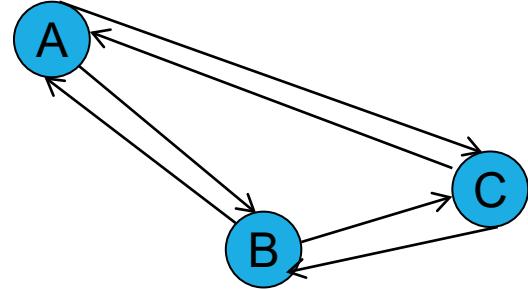
(again, no self ties)

e.g., if $N = 3$:

$$\#\text{possible edges} = (3*2)/2 = 3$$



EXAMPLE: (NUMBER OF EDGES):

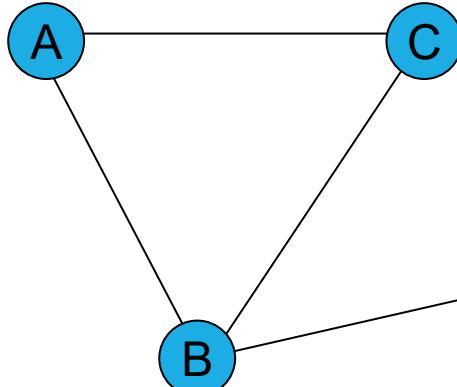


If there are 10 nodes in an undirected social network,
how many *possible* edges can exist?

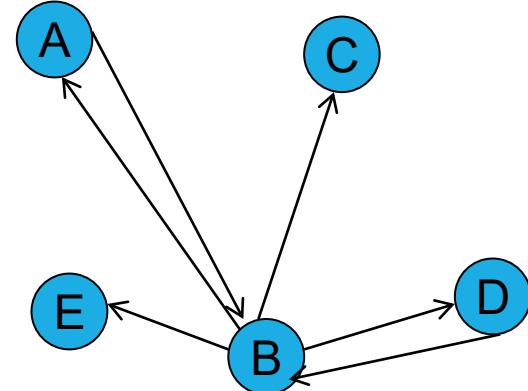
If there are 45 nodes in a directed social network,
how many *possible* edges can exist?

EXAMPLE:

Find the (edge) density of each of the following social networks:



Facebook connections



Twitter connections

3.2. CLUSTERING COEFFICIENT: CC

For a node v , calculate the *local* clustering coefficient value, $CC(v)$, of v as

the ratio of number of the edges between neighbours of the node v to number of possible edges between neighbours of the node v

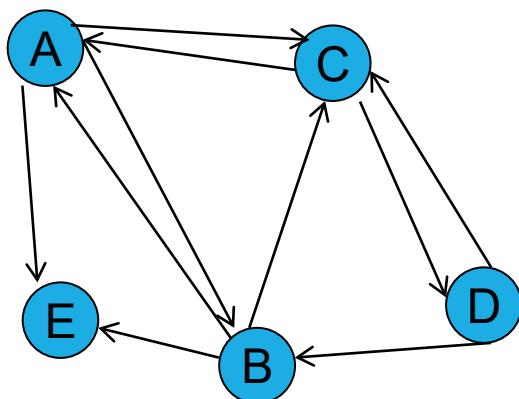
Same formula as before to calculate #possible edges for undirected and directed networks

In all cases: $0 \leq CC \leq 1$

Note: Only meaningful for nodes with at least 2 edges

NOTE: CLUSTERING COEFFICIENT WITH DIRECTED NETWORKS ...

As before but usually use the outdegree of a node
a measure of a node's neighbours, and use both
out and indegrees for the neighbour connections



CLUSTERING COEFFICIENT OF ZACHARY EXAMPLE NETWORK (CC1)

CC1 = 1 step

CC2 = 2 steps (friend-of-friends)



EXAMPLE:

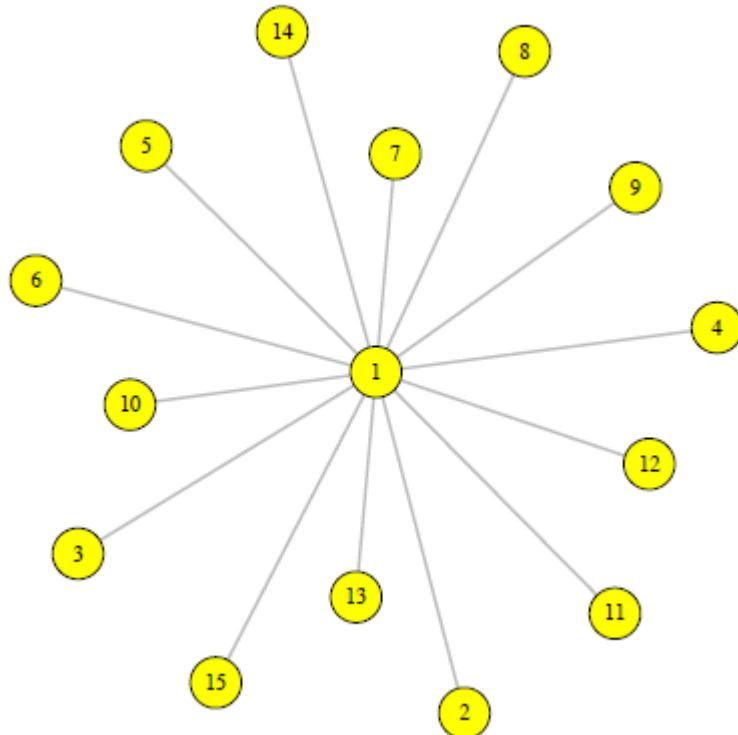
A social network of five users (represented by nodes 1, 2, 3, 4, 5), with **undirected** edges between the nodes, is represented by the following edge list:

(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (3, 4),(4, 5)

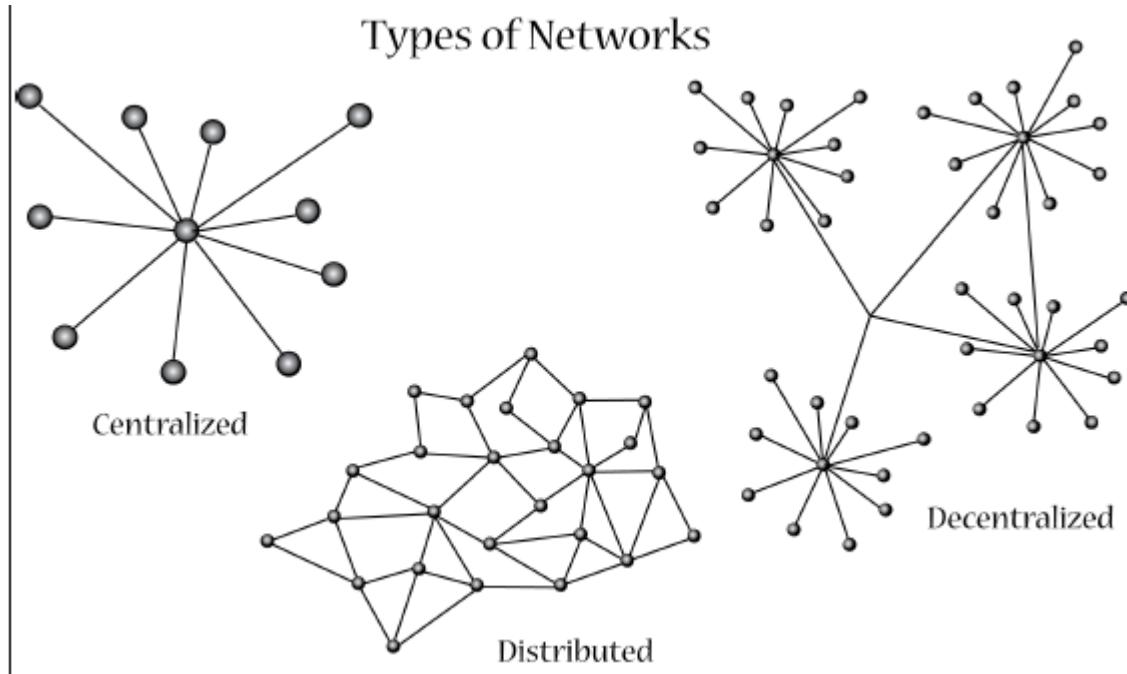
Find the clustering coefficient of node 1 in the network

Find the clustering coefficient of node 3 in the network

EXAMPLE: CLUSTERING COEFFICIENT FOR NODE 1 IN THIS (STAR) NETWORK

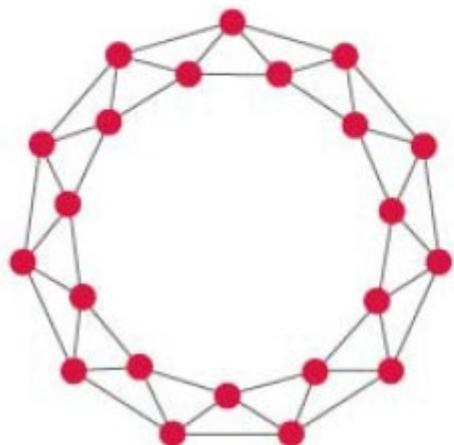


3.3 TYPES OF NETWORKS

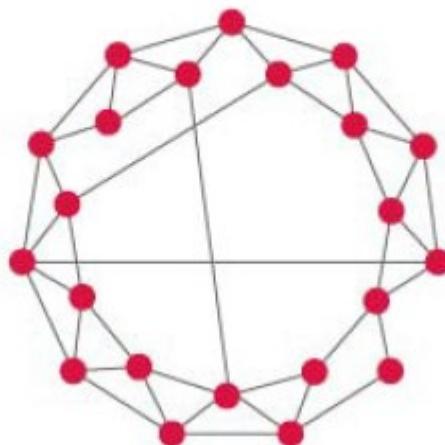


3.3 TYPES OF NETWORKS

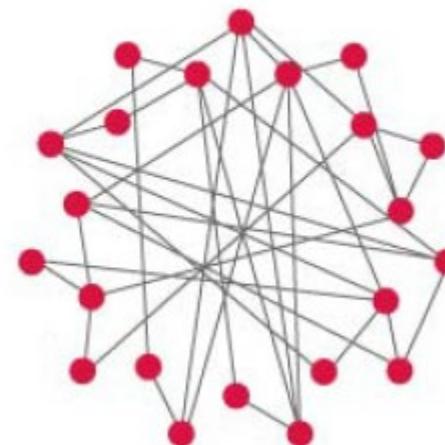
REGULAR NETWORK



SMALL-WORLD NETWORK



RANDOM NETWORK



DEFINITION: SMALL WORLD NETWORK



Celebrities

Amplify a Cause

Causes

Connect with Celebrities

You've probably heard of the Six Degrees concept. Any one person (including me, Kevin Bacon) is connected to any other person through six or fewer relationships, because it's a small world. SixDegrees.org is about using this idea to accomplish something good. **It's social networking with a social conscience.**



Most nodes in networks are not directly connected to each other, but most nodes can be reached from every other node by a small number of steps (for N nodes, approx. log N steps)

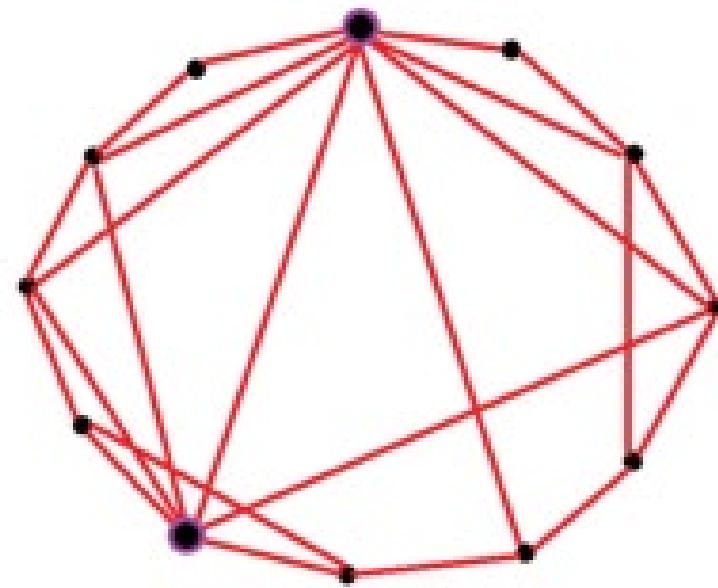
Many studies have shown that the small world phenomenon exists in many types of natural and artificial networks: brain neurons, websites, wikis, electric power grids, film actors in films, etc.

- e.g. Strangers linked by a mutual acquaintance

See: <http://www.sixdegrees.org/>

Example

- 12 nodes in the network
- Average degree = 3.83
- Average *shortest path length* between any 2 nodes: 1.803



SMALL WORLD NETWORK EXAMPLE FROM WIKIPEDIA

SUMMARY EXAMPLE

A social network of **six** users (represented by nodes A, B, C, D, E, F), with undirected edges between the nodes, is represented by the following edge list: (A, B), (B, C), (C,D), (C, E), (D,E), (D,F), (E,F)

Questions:

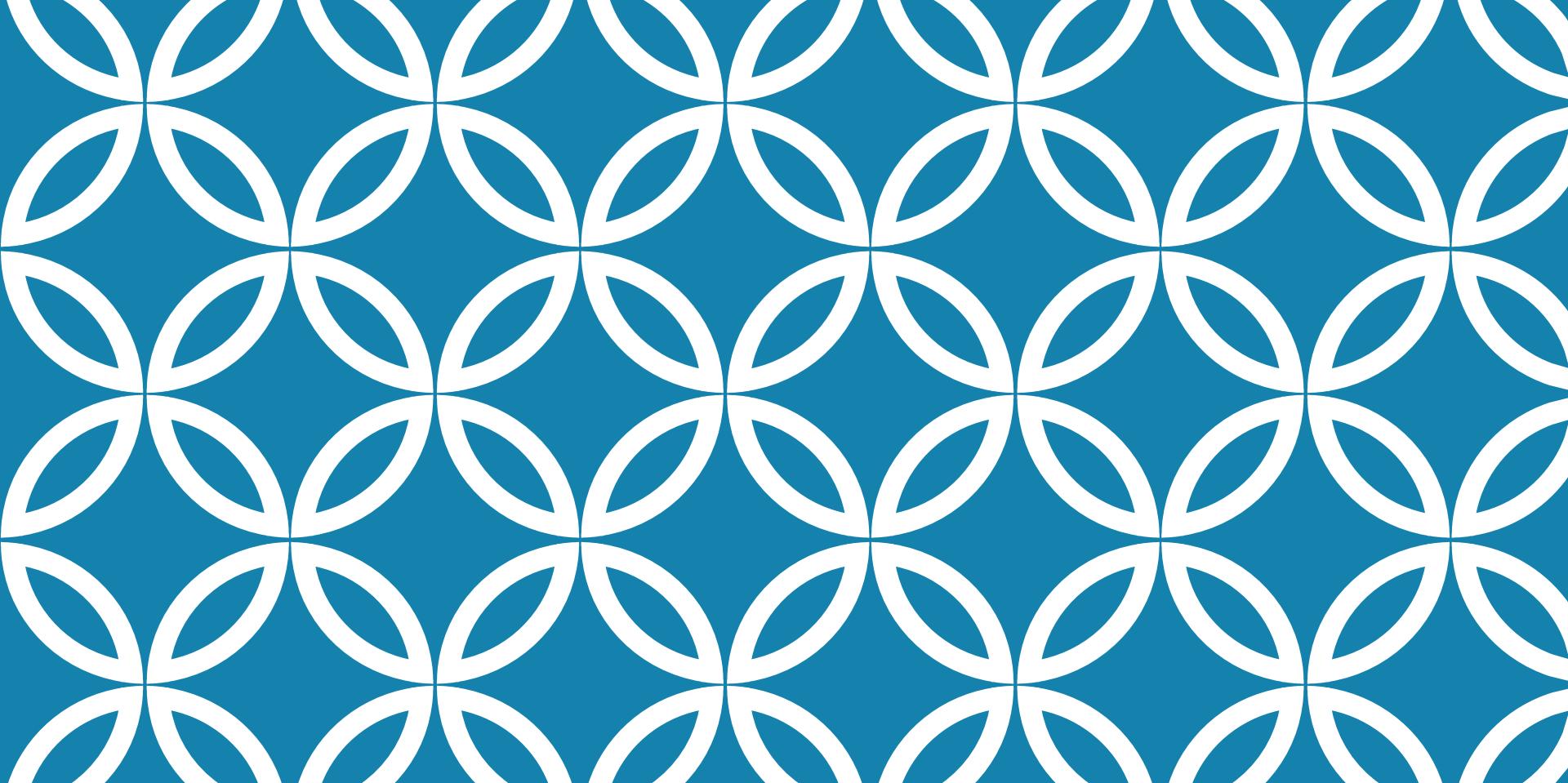
1. Draw the network and write out the adjacency matrix representation of the network
2. Find the average degree and find the node(s) that have a degree greater than the average.
3. List the path(s), and their length, that exist from node A to node E.
4. Define and calculate the local clustering coefficient value of the node D.
5. Define and calculate the (edge) density of the social network.

SUMMARY

Social Network Analysis is (another!) example of an area where our basic maths techniques (specifically matrix operations, graph algorithms) can find information important characteristics of our network in terms of:

- Node
- Relationships
- Network structure

These techniques are particularly applicable on Social Media Platforms



INTELLIGENT INFORMATION SYSTEMS VIA MACHINE LEARNING

CT102
Information
Systems

INTELLIGENCE

Difficult questions

- What is Intelligence?
- What is an Intelligent System?

Can be more specific by considering the goals of intelligent systems:

- Intelligent behaviour via computation

DEFINITION FROM THE BRITISH COMPUTER SOCIETY



Definition

Artificial intelligence can best be defined as the simulation of human intelligence, usually by computers. AI is very much an umbrella term for a good number of different approaches and theories. Some AI systems are good at solving one type of problem, such as playing chess or voice recognition. Present them with something they've never seen, and they'll stop. Other AI systems can teach themselves to solve new problems. Different systems have their strengths and weaknesses, and they all present their makers with different opportunities and even potential threats.

These technologies already play a significant role in improving the quality of life of all people in areas as essential as health care, transportation, communication and working conditions. It is hard to imagine a sector of society that will not be affected by AI.

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Many different approaches and techniques are used in the area of Artificial Intelligence.

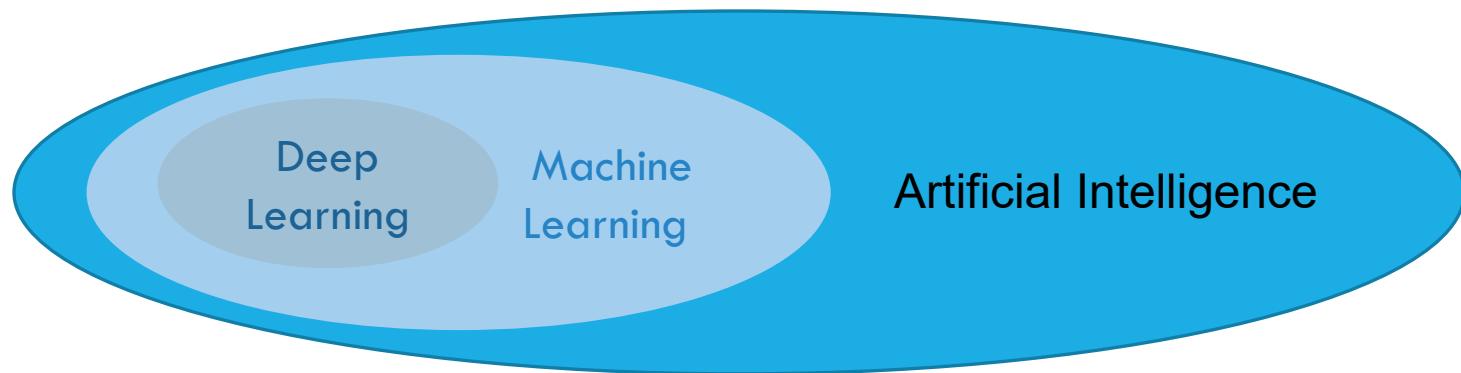
Many are fundamentally based on the concept of *learning*:

- i.e. for a system to be considered intelligent it should have the capacity to **learn**

Artificial Intelligence versus Machine Learning

Artificial Intelligence – automating intelligence

Machine Learning – learning patterns to create a model/algorithm automatically



WHY (get machines to) LEARN?

Many real-world problems are complex and it is difficult to exactly specify (algorithmically) how to solve the problem or what the solution is

Learning techniques are used in many domains to find solutions to problems that may **not** be obvious/clear to human users – especially when considering the large amounts of data that is now available online

FORMS OF ANIMAL AND HUMAN LEARNING



Environmental: Social, Language, Morals

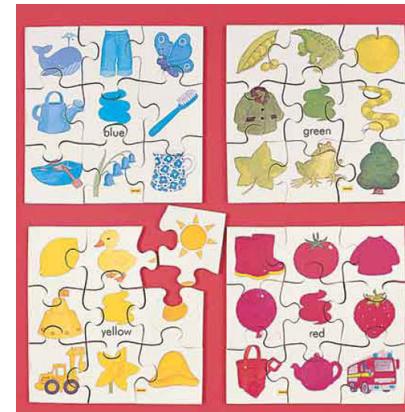
Play

Repetition or Rehearsal

Imitation

Formal/Informal/Non formal

Punishments/Rewards: Warnings, threats, promises



HOW IS “ROTE” LEARNING REPRESENTED IN INFORMATION SYSTEMS?

Rote learning: a memorisation technique based on repetition

In Computing this is achieved by explicit programming of rules

i.e. an algorithm can be considered an example of rote learning, e.g.

```
if (some set of circumstances occur)  
    perform a certain action
```

Can be considered to show some level of *intelligence* in the rules but does not have the capacity to *learn*

```
if (traffic light is red)  
    stop
```

HOW DOES MACHINE LEARNING WORK?

Concerned with the design and development of **programs** which **learn patterns** present in input (**training**) data

The learned patterns are used to make **predictions** about new **unseen (test)** data

LEARNING PATTERNS ...

- The patterns learned allow the program to generalise
- In order to generalise, the program creates a mathematical model
- The model contains the information on what pattern has been found
- Different types of machine learning programs learn different types of models
- The models they learn depend on what kind of data they have been given (training data)

PROBLEM AREAS WHERE MACHINE LEARNING IS APPLIED (1 OF 2)

Text Classification

Spam filtering (yes/no)

Recommendation

Speech Recognition (Speech to Text and Text to Speech)

Prediction

PROBLEM AREAS WHERE MACHINE LEARNING IS APPLIED (2 OF 2)

Object Recognition

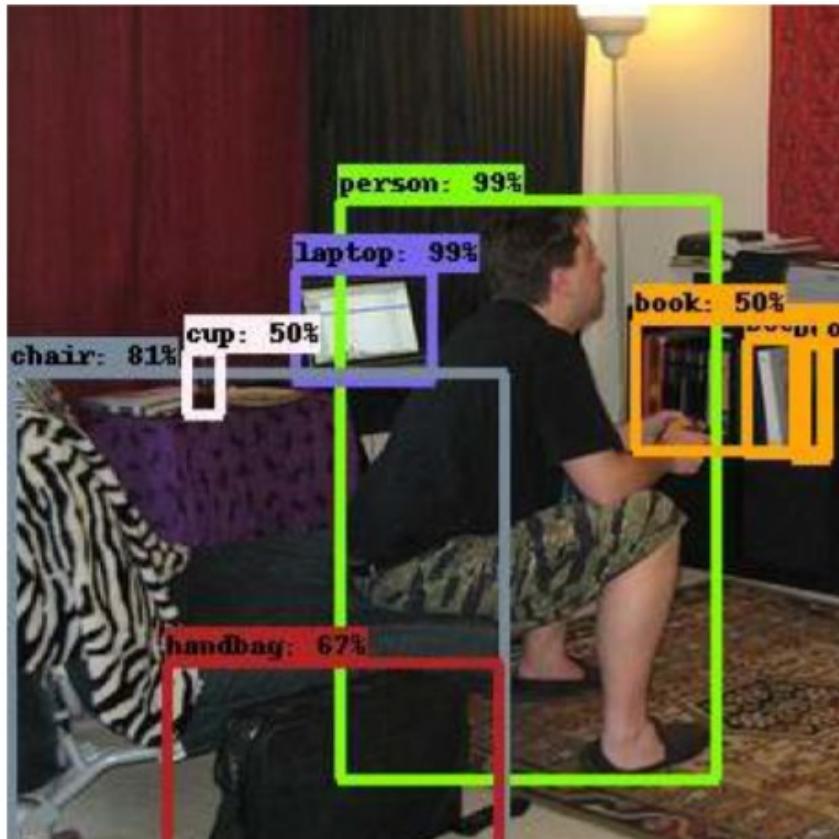
Pattern Recognition (e.g., Fraud detection, Medical applications)

Facial detection and Tagging

Sentiment Analysis

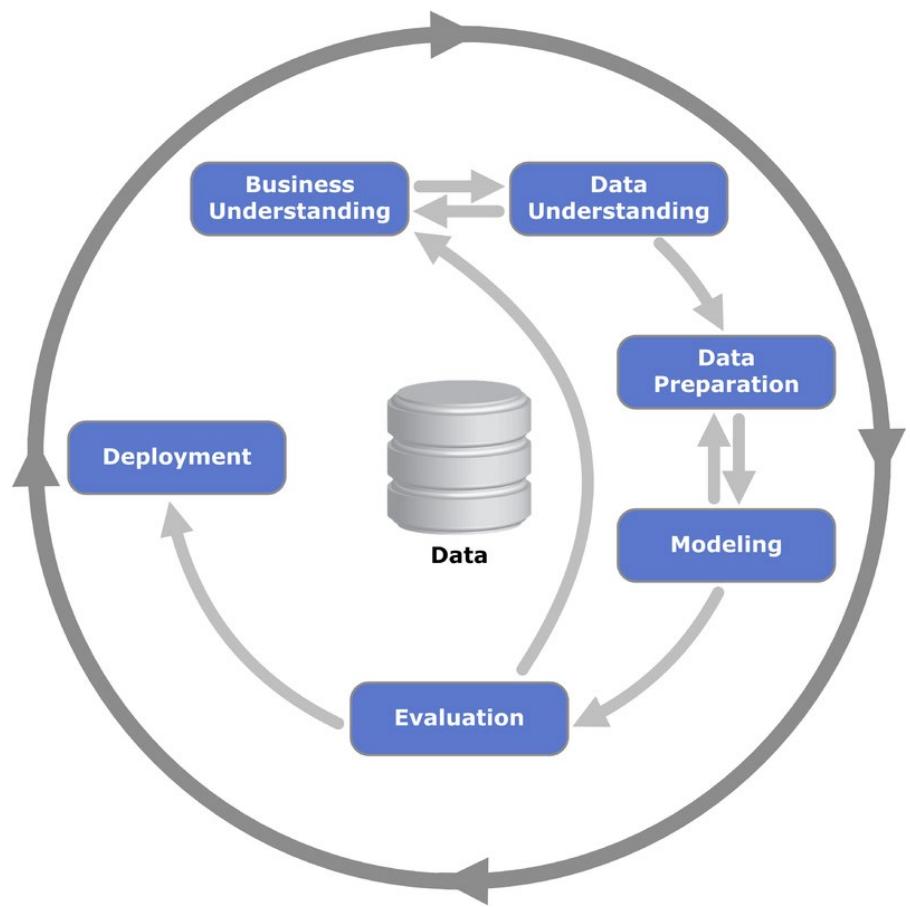
Computer Games

EXAMPLE: Identifying objects in a room



(a)

SAMPLE MACHINE LEARNING METHODOLOGY:



SAMPLE INPUT DATA

Vectors

Strings

Lists

Tables and Sets

Matrices

Images

Voice

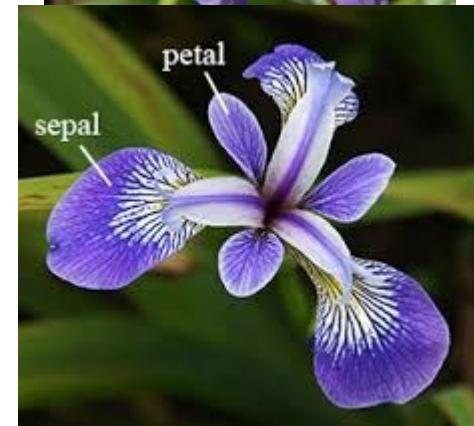
Video

EXAMPLE OF INPUT DATA:

Fisher's *Iris* data set

https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher's Iris Data				
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.2	<i>I. setosa</i>
5.4	3.9	1.7	0.4	<i>I. setosa</i>
4.6	3.4	1.4	0.3	<i>I. setosa</i>
5.0	3.4	1.5	0.2	<i>I. setosa</i>
4.4	2.9	1.4	0.2	<i>I. setosa</i>
4.9	3.1	1.5	0.1	<i>I. setosa</i>
5.4	3.7	1.5	0.2	<i>I. setosa</i>
4.8	3.4	1.6	0.2	<i>I. setosa</i>
4.8	3.0	1.4	0.1	<i>I. setosa</i>



TRAINING



Machine learning techniques require a **training stage** to learn some general rule or pattern (the mathematical model)

The training stage consists of a set of examples which can be:

- Labeled (Supervised training) “Learning by Example”
- Unlabelled (Unsupervised training) “Concept formulation”

Additional data may also be used to “**tune**” the model parameters (validation data) and **test** how good the learned model is (test data)

(MACHINE) LEARNING APPROACHES

Symbolic learning: knowledge is represented in the form of **symbolic descriptions** of the learned concepts, e.g., rules or hierarchies

Sub-symbolic learning: Knowledge is represented in a **sub-symbolic** form not readable by a user, e.g., in the structure, weights and biases of a trained neural network

Adaptive learning: “learning from and while interacting with the world”. e.g., reinforcement learning, evolutionary learning

SOME EXAMPLES

Symbolic learning:

- Decision trees and Decision Rules (for classification)
- Association Rules (for Market Basket Analysis).
- Nearest-neighbour approaches (for Clustering).

Sub-symbolic learning:

- Neural Networks and Deep learning (for classification, clustering, etc.).

Adaptive learning

- Reinforcement learning algorithms and Genetic Algorithms.

SUPERVISED LEARNING

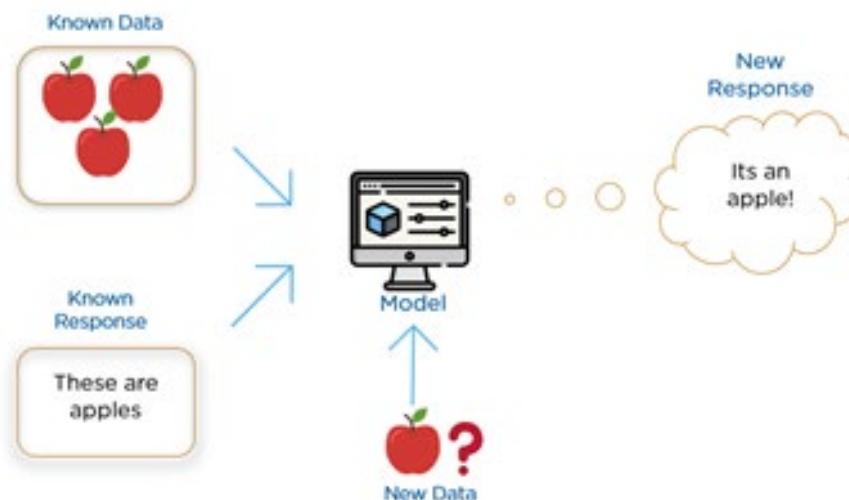


- The training data consists of a set of *training examples* or *instances*
- Each example is a pair consisting of an {input, output} pair where the input object is typically a vector and the output is the desired output value
- The output value is often called the *class*
- The parallel task in human and animal learning is often referred to as concept learning where an abstract or general idea is inferred or derived from specific instances

SUPERVISED LEARNING



The idea is: From the given training data learn patterns that will allow the machine learning technique to assign the correct output value to some unseen example



TRAINING (SUPERVISED)

The diagram illustrates the Fisher's Iris Data set, which consists of five features and one label/class. The features are Sepal length, Sepal width, Petal length, Petal width, and Species. The Species column serves as the label/class, indicating the three iris species: I. setosa, I. virginica, and I. versicolor.

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa
5.4	3.9	1.7	0.4	I. setosa
4.6	3.4	1.4	0.3	I. setosa
5.0	3.4	1.5	0.2	I. setosa
4.4	2.9	1.4	0.2	I. setosa
4.9	3.1	1.5	0.1	I. setosa
5.4	3.7	1.5	0.2	I. setosa
4.8	3.4	1.6	0.2	I. setosa
4.8	3.0	1.4	0.1	I. setosa

VALIDATION (SUPERVISED)

Rather than just having one phase of learning (with the training data) and one phase of testing the model learned (with the test data), **validation** is often used which uses a portion of the test data (**validation data**) to assess how well the model is being learned and to potentially improve the model as a result (“tuning” parameters of the model). Validation data can be used to help choose between a number of models (i.e. to pick the best).

Test data is still used as a final test of the model at the end of the learning phase

10-FOLD CROSS VALIDATION

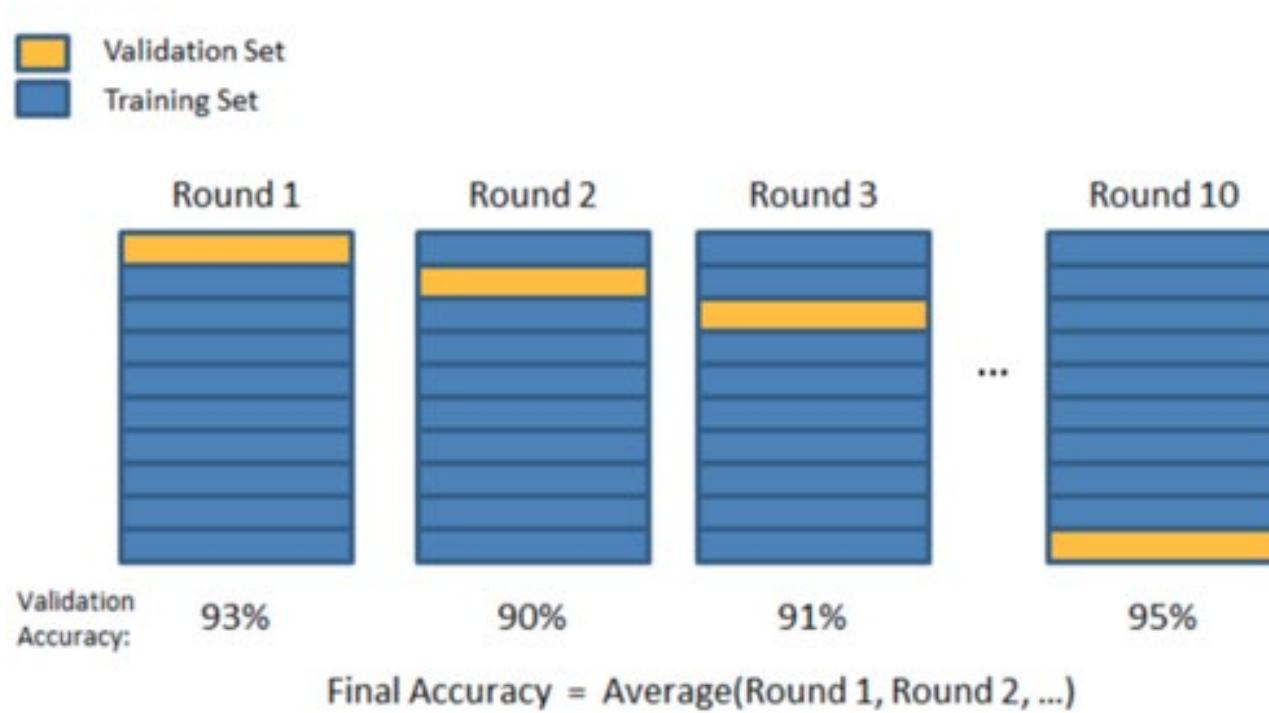
In **10-fold cross validation** the original dataset is randomly split 10 times into 10 equally-sized subsamples

From the 10 subsamples one subsample is used as the validation set to test the model and to fine-tune parameters; the remaining 9 subsamples are used to train the model

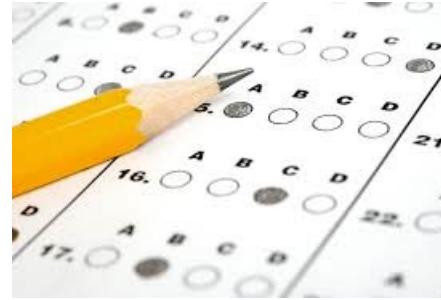
The process is repeated 10 times (10 folds) with each of the 10 subsamples used as the validation set exactly once

The 10 results are then averaged to create a single estimation of how well the model has learned

10-FOLD CROSS VALIDATION



TESTING (SUPERVISED)



Once the training and validation stage is complete the system must be **tested** with data not previously entered to the system for training or validation

This is to check if the rules/patterns learned only work for the examples already seen or if they can be generalised to other input examples

The “check” involves giving the system previously “unseen” data, and then comparing the system answer with the real answer

Generally, the test set, validation set and train set are originally in one set and are split before training and testing occurs

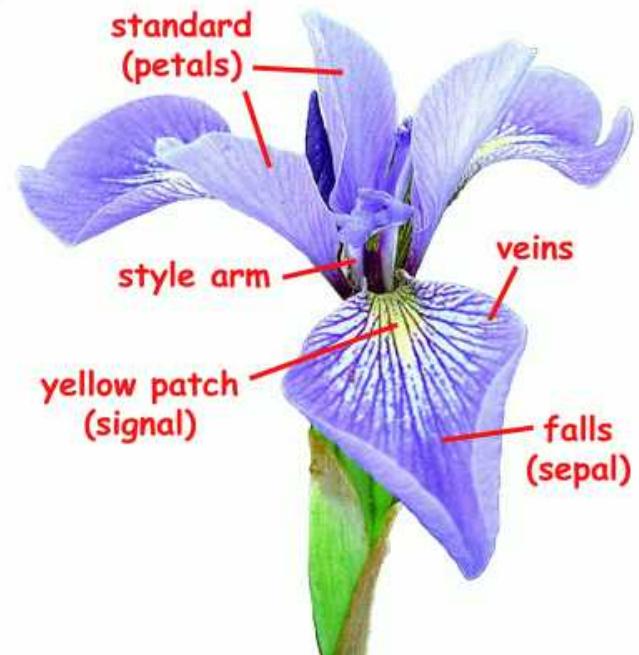
FOR EXAMPLE: IRIS DATA

The class is the species which has 3 possible values:

- Iris-setosa
- Iris-versicolor
- Iris-virginica

The Input object has 4 features

- Sepal length
- Sepal width
- Petal length
- Petal width



4 features

TRAINING

label/class

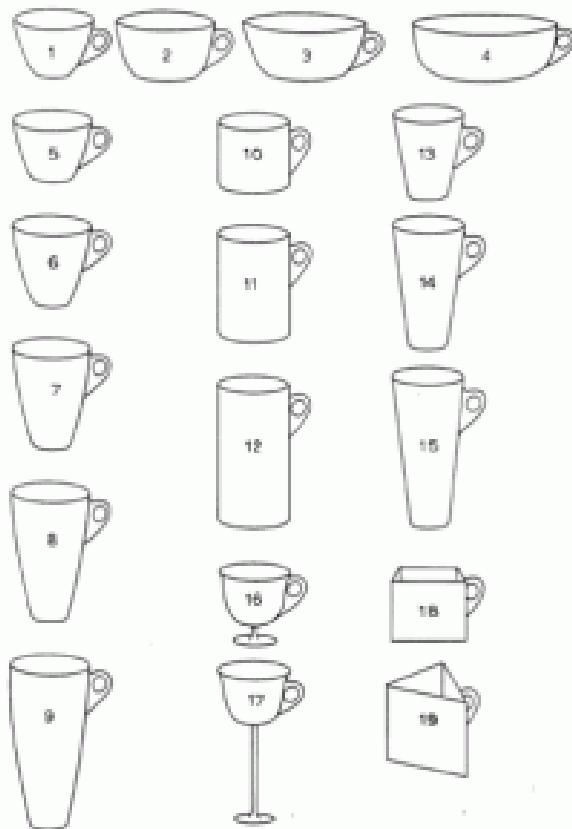
Fisher's Iris Data

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa
5.4	3.9	1.7	0.4	I. setosa
4.6	3.4	1.4	0.3	I. setosa
5.0	3.4	1.5	0.2	I. setosa
4.4	2.9	1.4	0.2	I. setosa
4.9	3.1	1.5	0.1	I. setosa
5.4	3.7	1.5	0.2	I. setosa
4.8	3.4	1.6	0.2	I. setosa
4.8	3.0	1.4	0.1	I. setosa

EXAMPLE: OBJECT RECOGNITION

Is it a cup or a bowl?

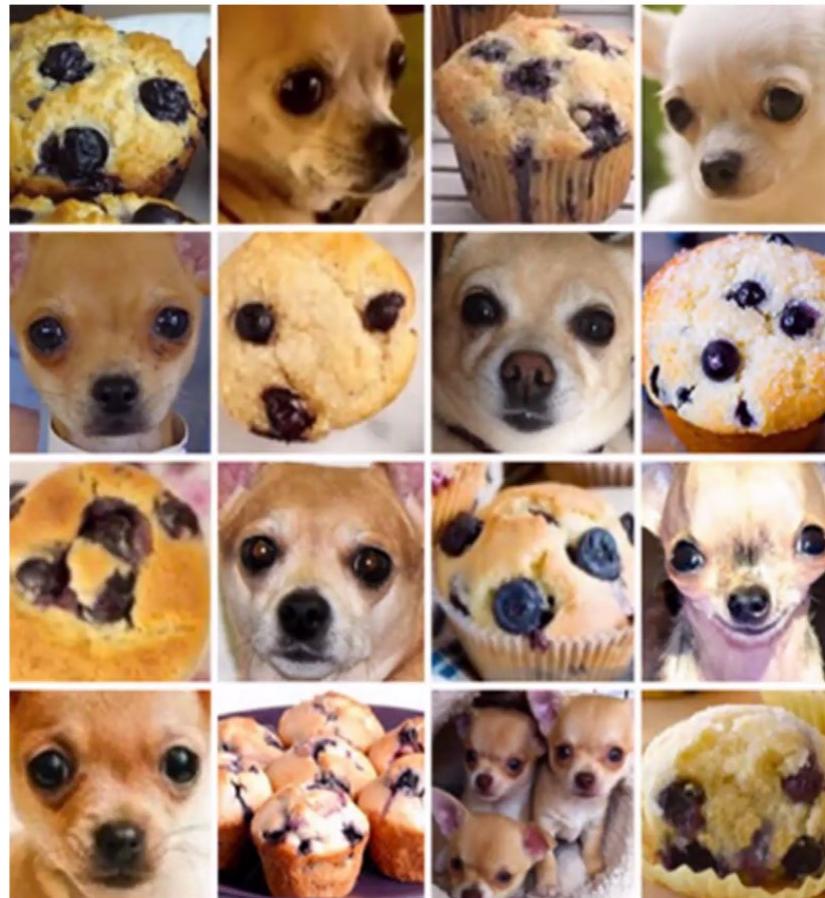
(“The boundaries of words and their meanings”, William Labov)



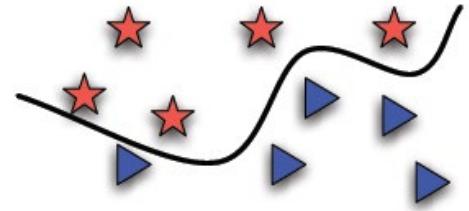
Some are easier to
classify than others

EXAMPLE: OBJECT RECOGNITION

Is it a dog or a muffin?



1. CLASSIFICATION



Classification is the process of finding the common properties among different entities and putting them into given *classes*.

Often results in *classification rules* which can be represented by a classification tree or decision tree

Classification is useful if the categories are known in advance

Classification is a classic example of a *supervised learning* problem

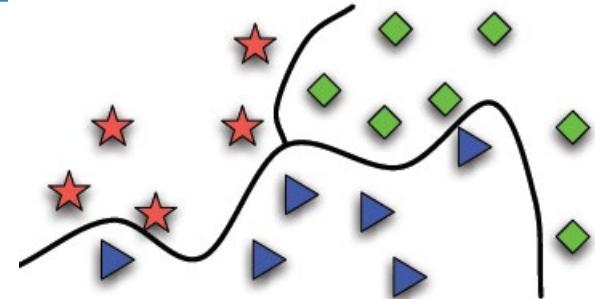
BINARY CLASSIFICATION



- There are only two classes (generally 1 and 0) so one output node is only needed.
- For example, given images of apples and other fruit we want a system to identify the images of apples
 - Can do this by associating 1 with apples, and 0 with “non-apples” for example.
- For example, given some text, want a system to identify if the text is fake or not
 - Can do this by associating 1 with fake and 0 with real for example

MULTICLASS CLASSIFICATION

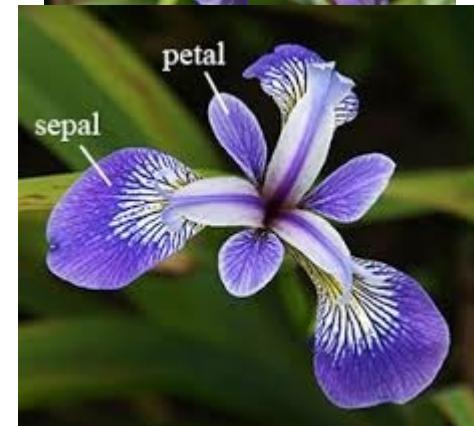
- An extension of Binary Classification where there may be a number of different classes
- For example, for some sample sentence classify if it is written in English, Irish, French, Hindi, Polish, Spanish (6 classes)
- For example, for the Iris dataset identify if the iris is one of three types (Iris-setosa, Iris-versicolor, Iris-virginica)



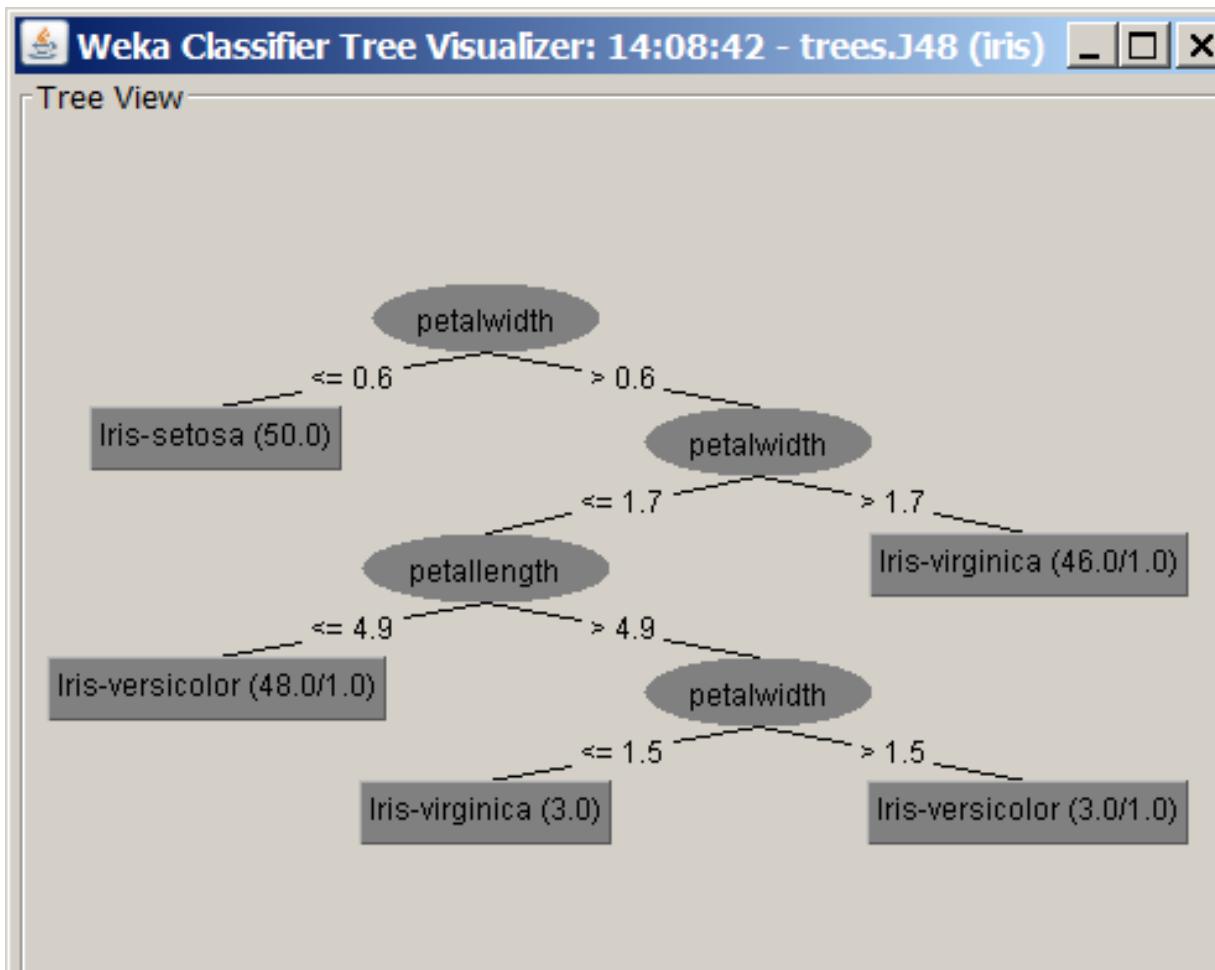
Fisher's Iris Data				
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa
5.4	3.9	1.7	0.4	I. setosa
4.6	3.4	1.4	0.3	I. setosa
5.0	3.4	1.5	0.2	I. setosa
4.4	2.9	1.4	0.2	I. setosa
4.9	3.1	1.5	0.1	I. setosa
5.4	3.7	1.5	0.2	I. setosa
4.8	3.4	1.6	0.2	I. setosa
4.8	3.0	1.4	0.1	I. setosa

Visualising the rules learned for the Iris classification example

Fisher's Iris Data				
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.2	<i>I. setosa</i>
5.4	3.9	1.7	0.4	<i>I. setosa</i>
4.6	3.4	1.4	0.3	<i>I. setosa</i>
5.0	3.4	1.5	0.2	<i>I. setosa</i>
4.4	2.9	1.4	0.2	<i>I. setosa</i>
4.9	3.1	1.5	0.1	<i>I. setosa</i>
5.4	3.7	1.5	0.2	<i>I. setosa</i>
4.8	3.4	1.6	0.2	<i>I. setosa</i>
4.8	3.0	1.4	0.1	<i>I. setosa</i>

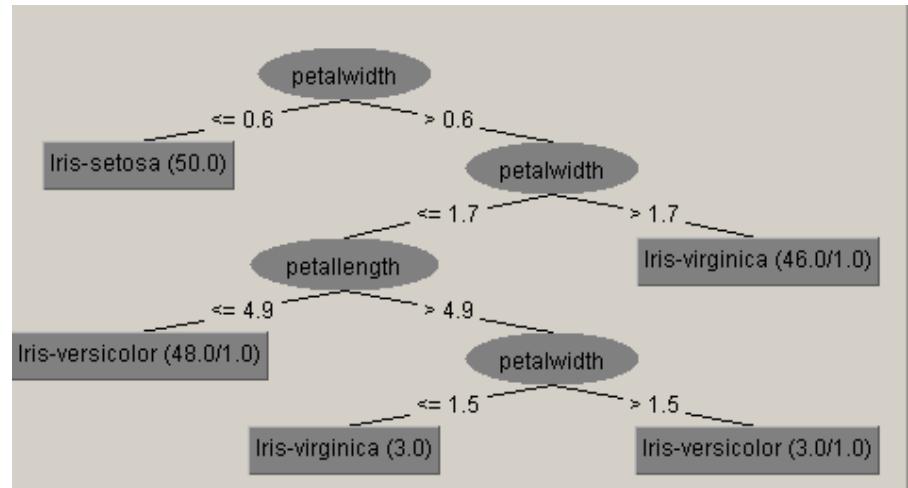


Visualising the rules learned for the Iris classification example



What do the rules look like?

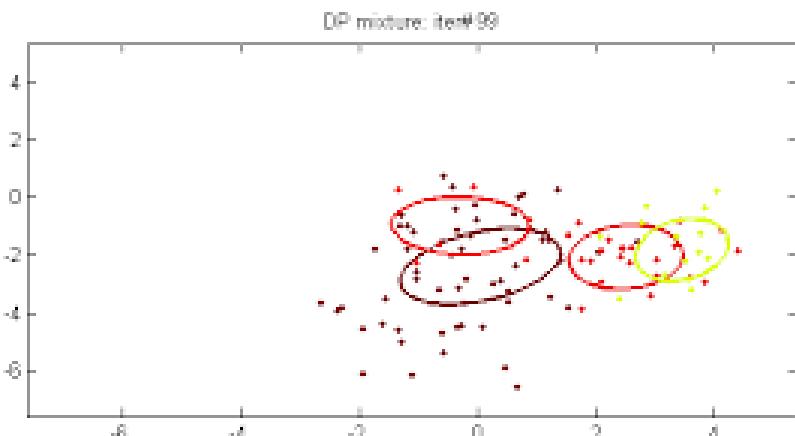
```
if (petalwidth <= 0.6)
    class= 'Iris-setosa';
else if (petalwidth <= 1.7) {
    if(petallength <= 4.9)
        class= 'Iris-versicolor';
    else if (petalwidth <= 1.5)
        class= 'Iris-virginica';
    else if (petalwidth > 1.5)
        class= 'Iris-versicolor';
}
else if (petalwidth > 1.7)
    class= 'Iris-virginica';
```



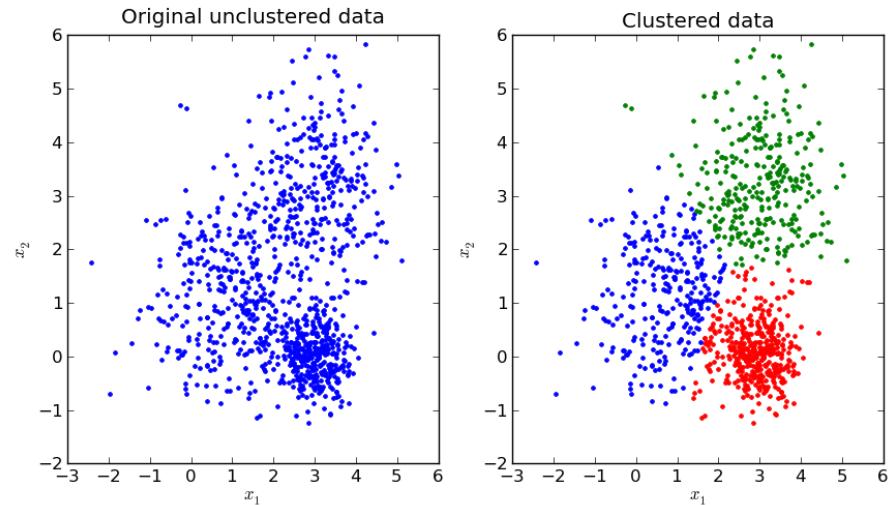
UNSUPERVISED LEARNING

- The training data consists of a set of *training examples* (input objects typically represented as a vector) but there is **no** output class associated with each input object

The idea is: find the objects which are most similar to each other



2. CLUSTERING



Given a large set of items, clustering partitions the documents into *clusters* such that the similarity of items within clusters is maximised and the similarity of items in different clusters is minimised.

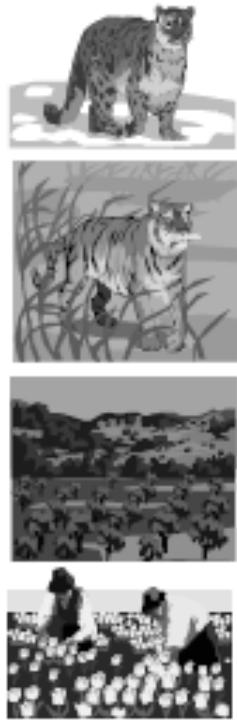
Clustering is used when categories are not known and must be found and items assigned to them.

The fundamental concept is to define some notion of *similarity* between items.

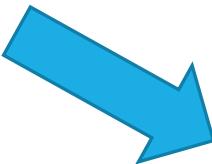
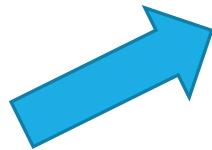
Clustering is an example of an *unsupervised learning* problem.

EXAMPLE:

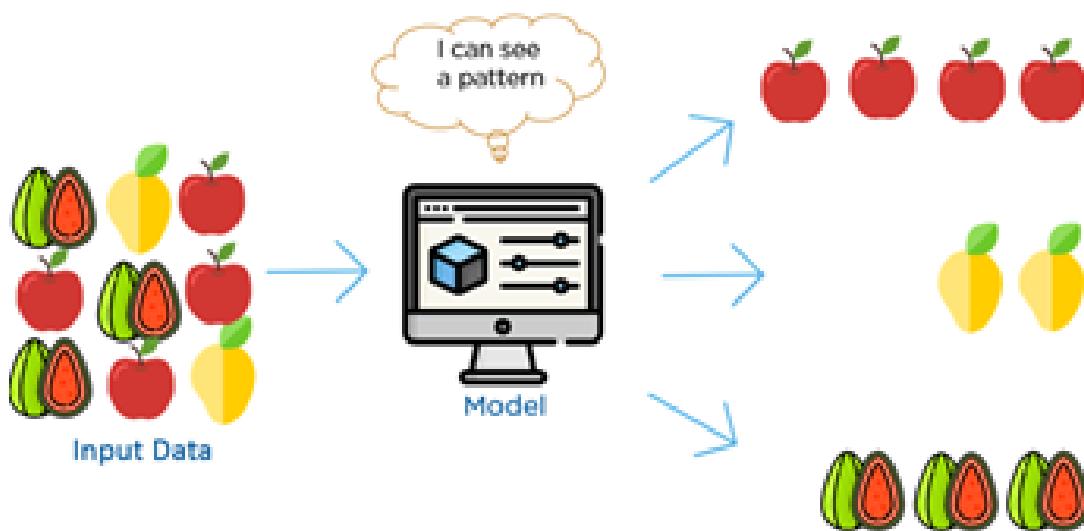
Unsupervised approach to image recognition



Images



FRUIT AGAIN ... but this time clustering them



Contrasting this with supervised approach to image recognition

mite	container ship	motor scooter	leopard
mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat

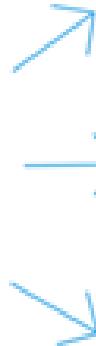
(from MNIST database)

RECOMMENDER SYSTEMS:

each dot on right hand side represents a movie
where the colour and closeness to each other
indicates similarity



I can see
a pattern

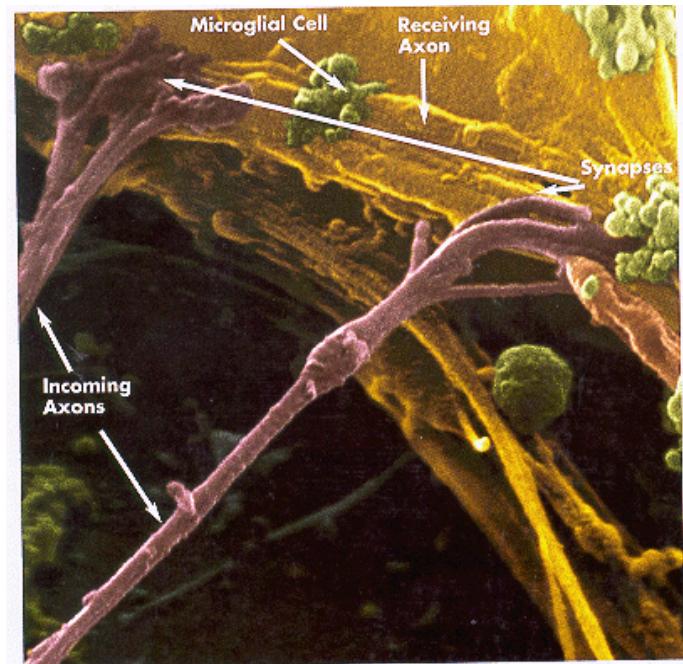
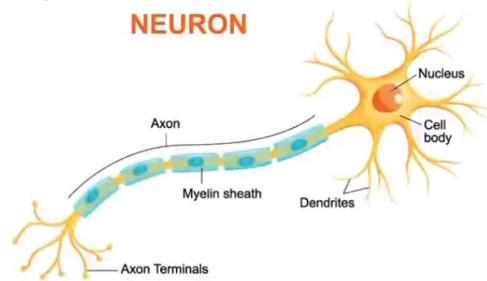


A SUB-SYMBOLIC APPROACH:

3. ARTIFICIAL NEURAL NETWORKS

- Inspired by simplified (known) workings of brain
- Neurons connected via links with each neuron possessing limited processing ability
- No single “unit of computation”
- Can be used for classification and clustering tasks

Biological
Neural Networks

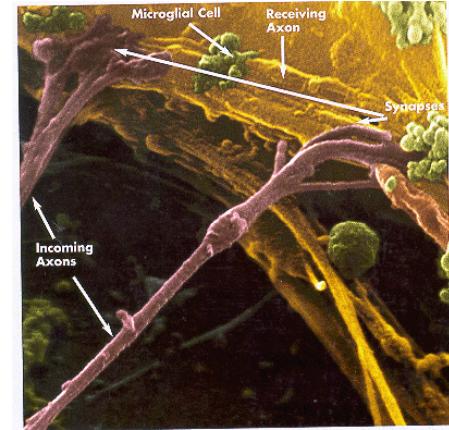


AN (ARTIFICIAL) NEURAL NETWORK CONSISTS OF:

Set of nodes, arranged as part of an:

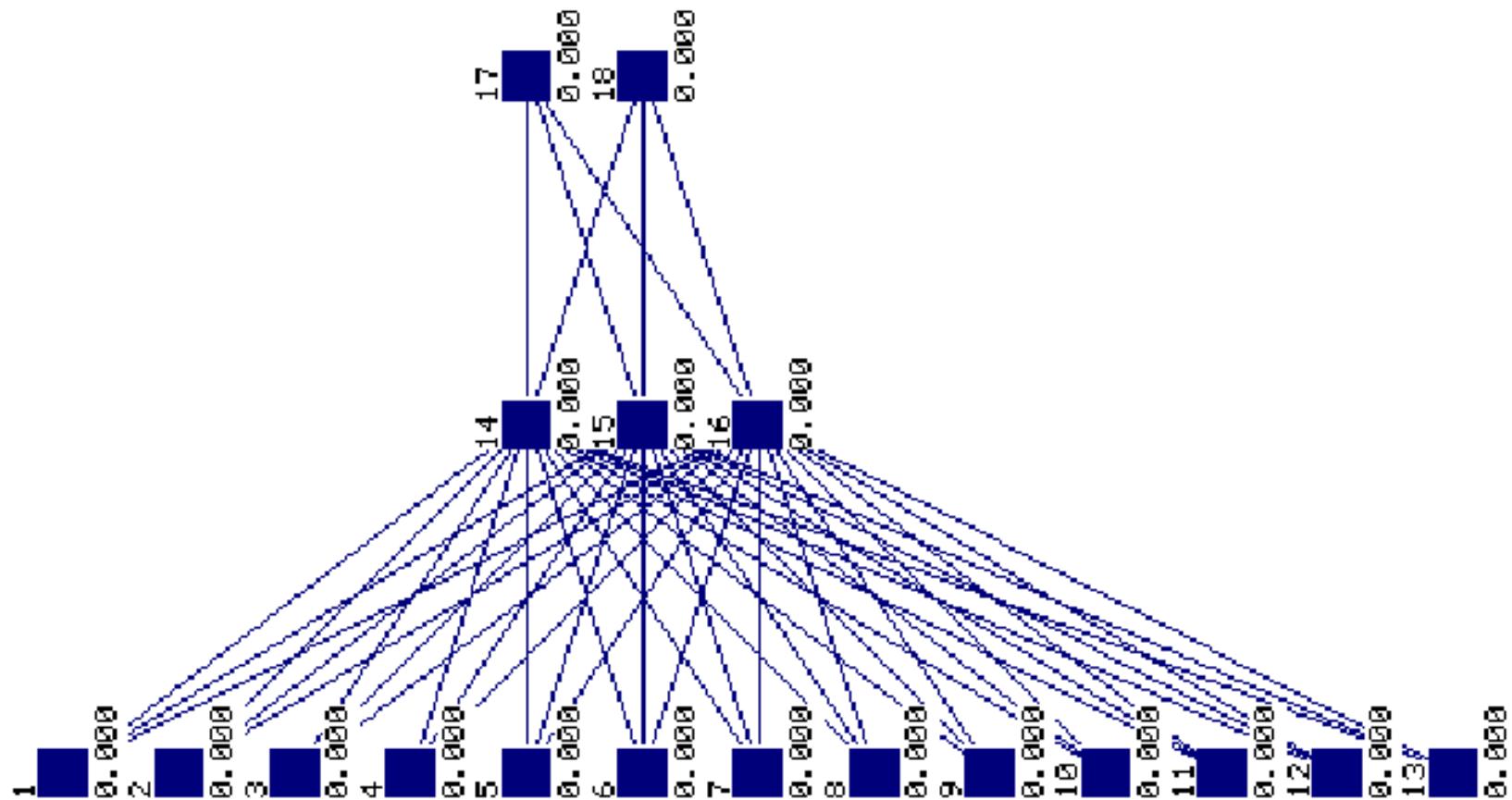
- input layer
- output layer
- one or more hidden layers between input and output layers

The input layer takes instances of the training data, the output layer produces result



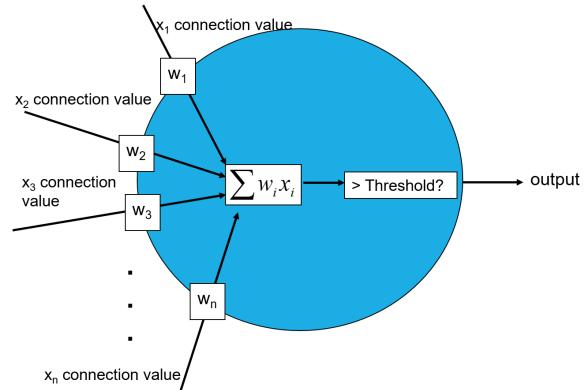
A SIMPLE NEURAL NETWORK:

13 input nodes, 3 nodes in inner layer and
2 output nodes

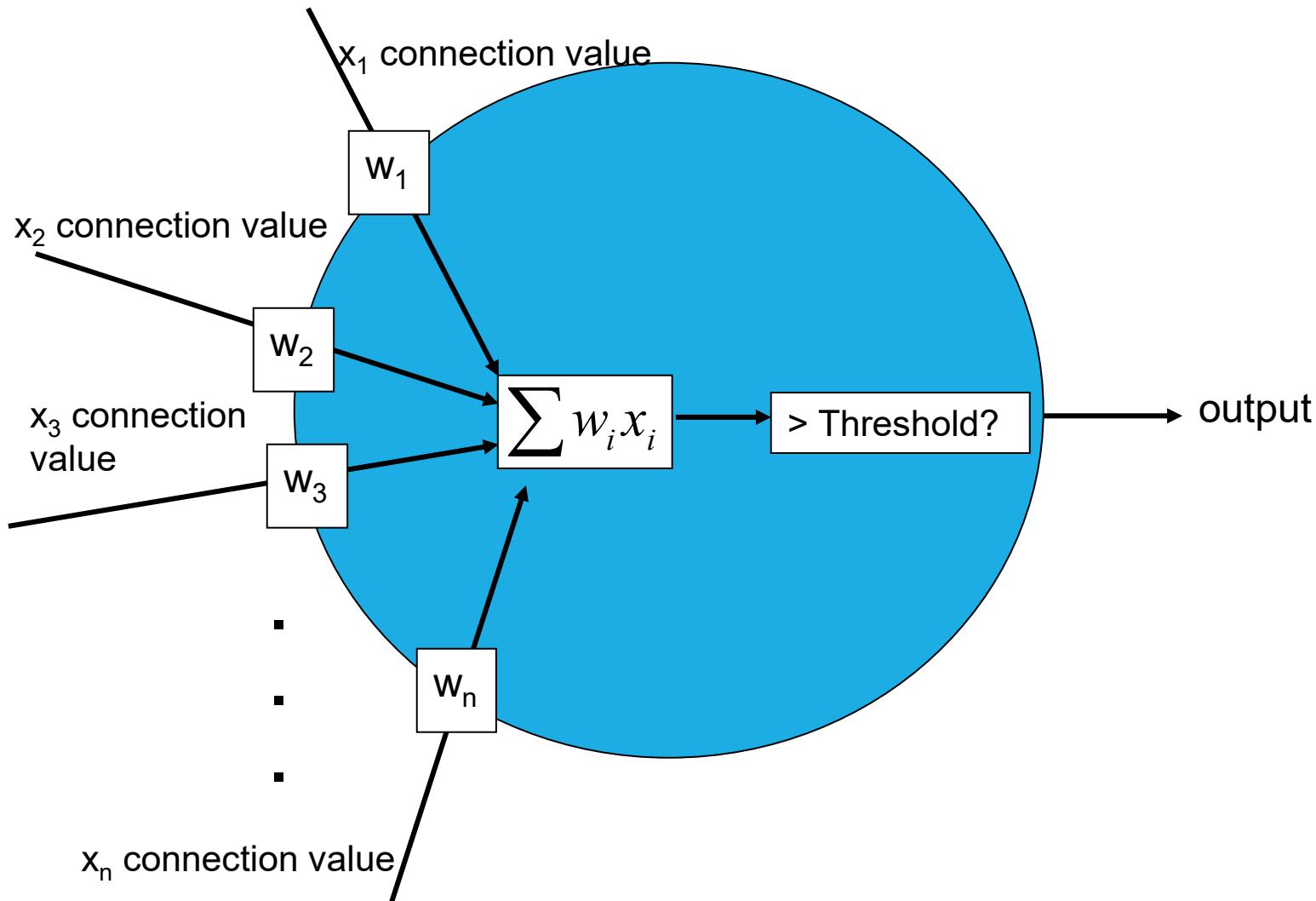


AN (ARTIFICIAL) NODE COMPRISSES:

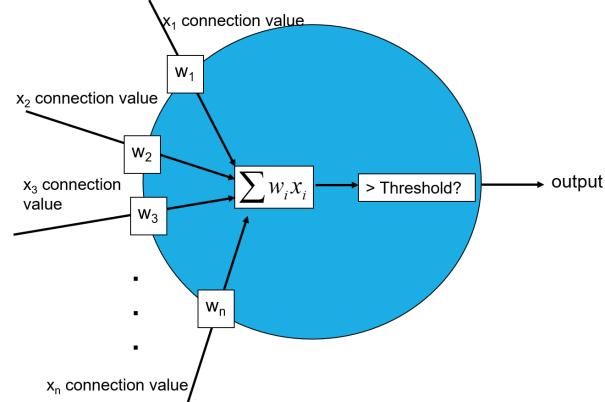
- One or more **Input** Values (coming from other nodes)
- One or more **Weights** associated with each input
- One **Output** Value
- An **Activation Function**: which is some summation of the weights and values, giving an *activation value*
- A **Threshold** value



SAMPLE STRUCTURE OF A NODE



THRESHOLD VALUE



A value will only be output from the node if the *activation value* is greater than the *threshold value*

The output of one neuron acts as an input to another neuron

HOW DOES LEARNING TAKE PLACE?

For example, given a classification task and labelled training data:

In the output layer, the actual output value, O_a is compared with the desired output value O_d

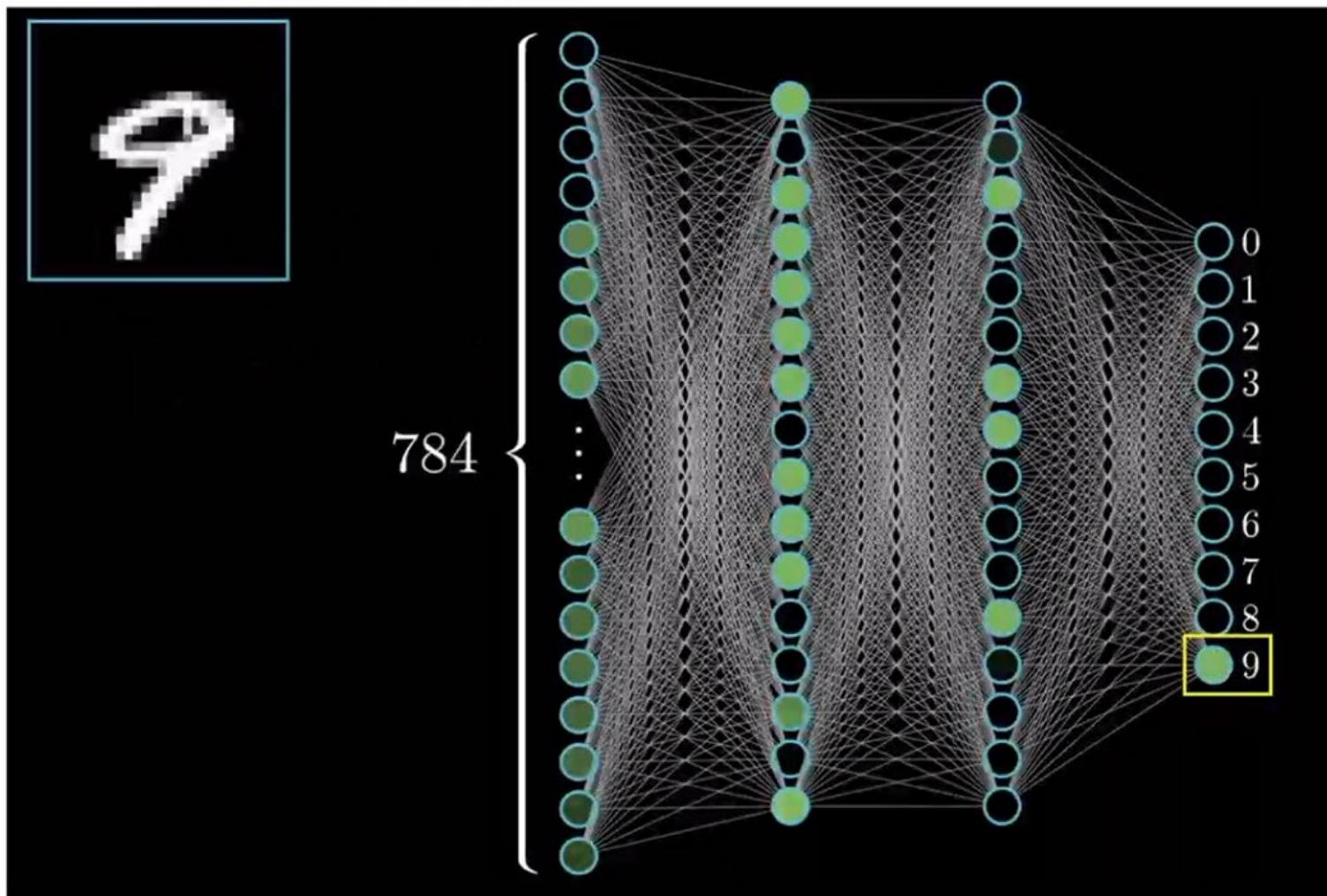
A degree of error can be calculated by:

$$O_d - O_a$$

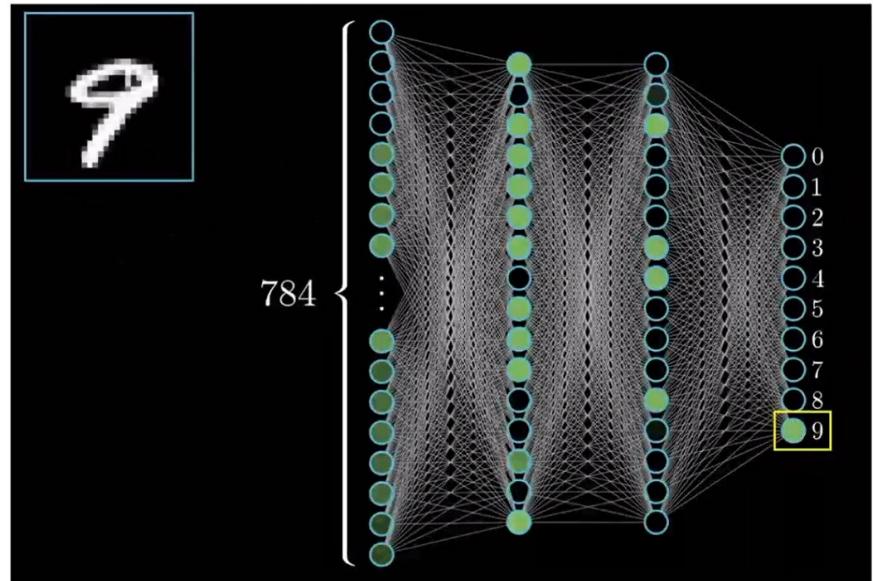
The error is “back-propagated” throughout the network by modifying weights according to the degree of error

Example:

Recognising hand-written numbers (images)



Recognising hand-written numbers



28x28 pixel image so input is 784 values which are either “on” (1/white) or “off” (0/black).

The output consists of 10 nodes and only one will be “on” (value of 1) to indicate which number has been recognised (9 in the above example)

DEEP LEARNING

- Deep learning is a new area within machine learning which uses neural networks using multiple layers
- Has had good successes at hard problems such as image recognition, speech and audio recognition, language processing and bioinformatics

CHARACTERISTICS OF DEEP LEARNING APPROACHES

- One major difference to other learning approaches is with respect to how the training instances/objects are represented
- Deep learning attempts to find good representations itself by learning the representations using unlabelled data:
Specifically higher level representations are derived from lower level features /representations

EXAMPLE OF SUCCESS OF DEEP LEARNING APPROACHES

MAINSTREAM VOICE RECOGNITION

- Large-scale automatic speech recognition is the first and most convincing successful case of deep learning, embraced by both industry and academia
- All major commercial speech recognition systems - Microsoft's Cortana, Xbox, Skype Translator, Google Now, Apple Siri, Baidu and iFlyTek voice search - and others are based on deep learning approaches

EXAMPLE OF SUCCESS OF DEEP LEARNING APPROACHES OBJECT RECOGNITION

The Deep learning research team at Google (Google Brain Project) created a neural network that learned to recognize higher-level concepts, such as cats, only from watching unlabelled images taken from YouTube videos

- Created a neural network with 16,000 computer processors as the nodes and over 1 billion connections between these nodes
- Presented the network with 10 million digital YouTube thumbnail images found in YouTube videos (unlabelled)

Reference: Ng, Andrew; Dean, Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning". Proc of the 29th Conf. on Machine Learning



An image of a cat that a neural network taught itself to recognize. Jim Wilson/The New York Times

John Markoff (25 June 2012). "How Many Computers to Identify a Cat? 16,000." New York Times.



EXAMPLE OF SUCCESS OF DEEP LEARNING APPROACHES DRIVERLESS CARS

- Self-driving cars are not programmed in the classical sense of mapping human driving decisions to a set of IF-THEN rules except for very simple rules such as:

IF trafficLightInput == red **THEN** stop

- Most of the decisions must be learned based on training data comprising of a huge number of traffic situations
- For example, Google has driven almost two million kilometers on public roads with test drivers

GENERAL ETHICAL CONCERNS

- Accuracy is extremely important ... but not enough
- Responsibility – “with power comes responsibility”
- Explainability – why certain actions were taken; why a certain result was given – *understanding*
- Auditability – trace back actions that led to an outcome - *retracing, replicating*
- Fairness – biases in data can lead to biases (thus unfairness) in results

ACCURACY & ROBUSTNESS

What happens if an image is modified slightly?



\mathbf{x}
“panda”
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$
“nematode”
8.2% confidence

=



$\mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$
“gibbon”
99.3 % confidence

In J. Goodfellow, Jonathon Shlens & Christian Szegedy (2014). Explaining and harnessing adversarial examples. arXiv:1412.6572
Traffic Sign Examples Image Credit to Jiefeng Chen & Xi Wu (2019). <https://www.altacognita.com/robust-attribution>

Human: Sees panda in both images

ML Model: Classifies image without perturbation (on left) as “Panda” with 57.7% confidence; classifies image after perturbation as “Gibbon” with 99.3% confidence.

ACCURACY & ROBUSTNESS

What happens if an image is modified slightly?



classified as
Stop Sign

+ .007 ×



$\text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$
“nematode”
8.2% confidence

=



classified as
Max Speed 100

Human: Sees stop sign in both situations

ML Model: Classifies image without perturbation (on left) as “Stop”; classifies image after perturbation as “Max Speed 100”

BIASED TRAINING AND TEST DATA

Machine learning techniques can only generalise based on the data that has been given as input

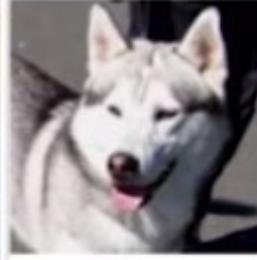
“Poor” biased data may cause problems such as:

- May not generalise to other very similar patterns.
- May over generalise.
- Note that if test data is being used, but also shows the same bias as the training data, then accuracy can appear to be very good

Example 1

a Machine Learning system to distinguish between images of huskies and wolves which is quite a difficult task ...

Only 1
mistake!

	Predicted: wolf True: wolf		Predicted: husky True: husky		Predicted: wolf True: wolf
	Predicted: wolf True: husky		Predicted: husky True: husky		Predicted: wolf True: wolf

But if we can analyse what is being used to make these decisions



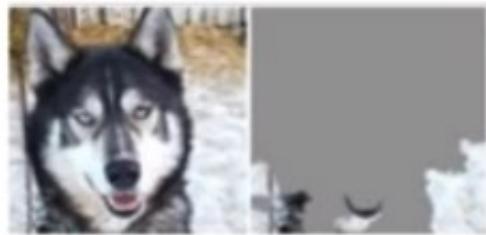
Predicted: **wolf**
True: **wolf**



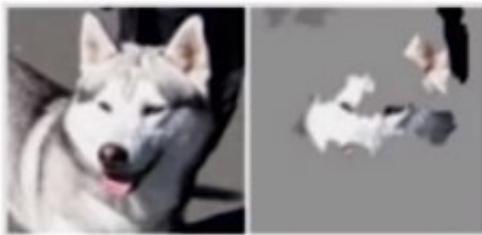
Predicted: **husky**
True: **husky**



Predicted: **wolf**
True: **wolf**



Predicted: **wolf**
True: **husky**



Predicted: **husky**
True: **husky**



Predicted: **wolf**
True: **wolf**

EXAMPLE 2: FACIAL RECOGNITION SOFTWARE



In 2018, a study found that three gender-recognition AI programs using images of faces (from IBM, Microsoft and a Chinese company called Megvii) correctly identify a person's gender 99% of the time ... if the person was a white man.

For Asian and African-American women, it was only correct 35% of the time (aka wrong 65% of the time!)

The reason was shown to be the bias in the training data which contained a much higher number of examples of white men than women and than Asian and African-American women and so the machine learning model had more training on the white men and little chance to generalise (or learn) from other images.

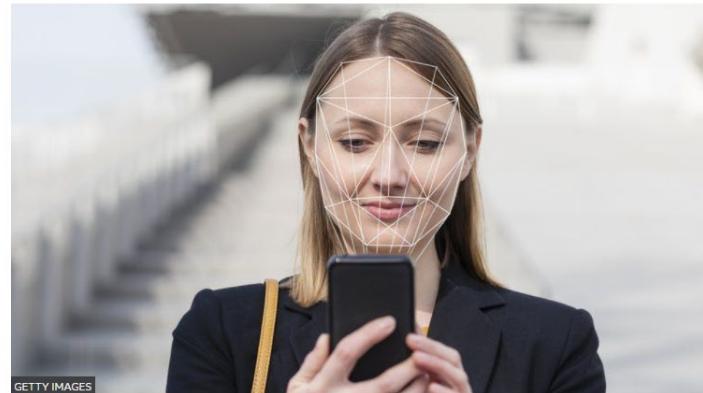
All companies said they re-trained their systems on new data.

OTHER EXAMPLES IN THE NEWS ...

Facebook to end use of facial recognition software

By Beth Timmins
Business reporter, BBC News

© 2 November



Facebook has announced it will no longer use facial recognition software to identify faces in photographs and videos.

There have been growing concerns about the ethics of facial recognition technology, with questions raised over privacy, racial bias, and accuracy.

Regulators had not yet provided a clear set of rules over how it should be used, the company said.

It has faced a barrage of criticism over its impact on its users.

Until now, users of the social media app could choose to opt in to the feature which would scan their face in pictures and notify them if someone else on the platform had posted a picture of them.

In a blog post, Jerome Pesenti, vice president of artificial intelligence at the firm said: "Amid this ongoing uncertainty, we believe that limiting the use of facial recognition to a narrow set of use cases is appropriate."

One in three councils using algorithms to make welfare decisions

Amazon scraps secret AI recruiting tool that showed bias against women

Sarah Mars

✉ @sloumarsh

Tue 15 Oct 2019



Automation has been key to

Artificial intelligence
(AI)

Updated / Wednesday, 10 Oc

Henry McDonald

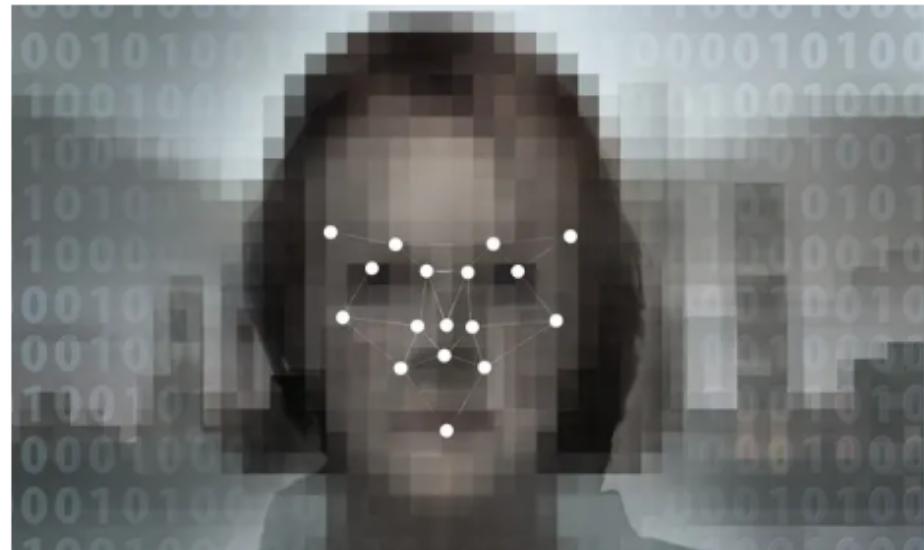
Thu 12 Dec 2019 14.07 GMT



137

AI expert calls for end to UK use of 'racially biased' algorithms

Prof Noel Sharkey says systems so infected with biases they cannot be trusted



▲ Facial recognition technology has also come under scrutiny. Photograph: Fanatic Studio/Gary Waters/Getty/Collection Mix: Subjects RF

An expert on artificial intelligence has called for all algorithms that make life-changing decisions - in areas from job applications to immigration into the UK - to be halted immediately.

Prof Noel Sharkey, who is also a leading figure in a global campaign against "killer robots", said algorithms were so "infected with biases" that their decision-making processes could not be fair or trusted.

Other Ethical concerns

Be afraid: The era of easy deepfake videos is upon us

Digitally altered audio-visual content puts us on the brink of an information apocalypse

Thu, Nov 14, 2019, 06:00

Marie Boran



from the Irish Times, Nov 2019

Artificial intelligence (AI)

New AI fake text generator may be too dangerous to release, say creators

The Elon Musk-backed nonprofit company OpenAI declines to release research publicly for fear of misuse

Alex Hern

@alexhern
Thu 14 Feb 2019 17.00 GMT



4,619 572



▲ The AI wrote a new passage of fiction set in China after being fed the opening line of Nineteen Eighty-Four by George Orwell (pictured). Photograph: Mondadori/Getty Images

The creators of a revolutionary AI system that can write news stories and works of fiction - dubbed "deepfakes for text" - have taken the unusual step of not releasing their research publicly, for fear of potential misuse.

OpenAI, a nonprofit research company backed by Elon Musk, Reid Hoffman, Sam Altman, and others, says its new AI model, called GPT2 is so good and the risk of malicious use so high that it is breaking from its normal practice of releasing the full research to the public in order to allow more time to discuss the ramifications of the technological breakthrough

Text-based Editing of Talking-head Video (SIGGRAPH 2019)

Watch later Share



SUMMARY

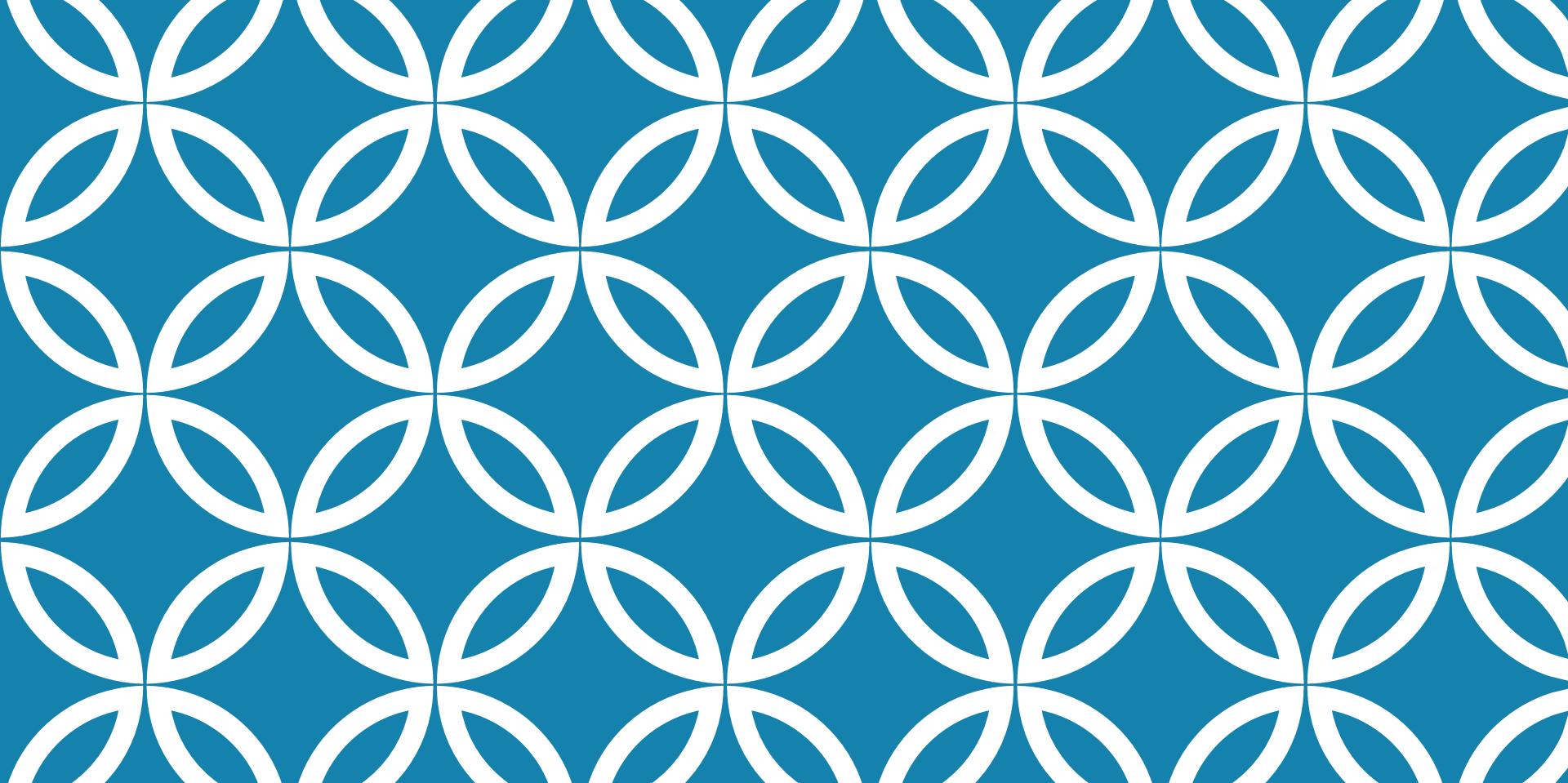
- Machine learning is now a very large area within Computing and Information Systems
- Where it was once associated with research projects only, in the last number of years, it is becoming mainstream
 - This is due to the availability of machine learning software and APIs, advances in deep learning and the advances in hardware.
 - Many ethical issues:

“The promise of AI is that it will imbue machines with the ability to spot patterns from data, and make decisions faster and better than humans do. What happens if they make worse decisions faster?”

(Guardian Editorial, September 2019)

IMPORTANT CONCEPTS

- What is machine learning and what problems does it solve?
- Symbolic Vs Sub-Symbolic Learning
- Supervised Vs Unsupervised Learning
- Training Data, Validation Data, Testing data
- Data Features and classes/labels
- Classification and Clustering
- Neural networks, Deep learning and its successes
- Bias in data and how Machine Learning models are affected



TOPIC: DATA & INFORMATION SYSTEMS SECURITY

DATA SECURITY



For as long as there has been data, there has been a security aspect to this data:

- From secret communication to secret passwords to digital identities

Many often-cited examples of famous secure codes, e.g., German Enigma machine



Much media coverage of **breaches** to data security and **penalties** for poor/inadequate security approaches ... government, banks, surveillance, cloud, social media, whistleblowers, DDoS attacks, etc.



DATA AND INFORMATION SECURITY

Involves **defending** data, information and information systems from unauthorised:

- access
- use
- disclosure
- disruption
- modification
- perusal/inspection
- recording
- destruction



www.bigstock.com · 2354900

INFORMATION SECURITY THREAT: HACKING

Hacking can be defined as any action that exploits a weakness in a computer system or network, often spreading malware as a result.

EXAMPLES OF DATA SECURITY BREACHES:

Eavesdropping/Monitoring/Surveillance: obtaining or viewing information without explicit authority

Masquerading: sending/receiving messages using other's identifier

Tampering: stealing messages and altering their contents

Replaying: storing messages and sending them at later date

Infiltrating: accessing system in order to run programs that implement an attack (virus, worm, Trojan horse, etc.)

Decoding: decoding encrypted messages

MALWARE: MALICIOUS SOFTWARE

Virus: software that is designed to copy itself on to the host system and replicate

Worm: a type of virus that uses flaws in the OS to spread itself and cause harm – often degrading performance in some way

Trojan Horse: usually an email attachment which looks legitimate but contains an executable program which can modify files, steal confidential data, encrypt data, spread itself using your contacts, prevent you controlling your own computer, etc.

Bots: an advanced form of worm that are designed to interact over the internet without the need for human interaction.

MALWARE: MALICIOUS SOFTWARE

Ransomware: program encrypts files on the computer where it runs and displays a message seeking payment before files will be decrypted and/or not published

Spyware: software which installs itself secretly and logs and reports information on user actions (e.g. keyboard strokes/passwords). Can be used as part of targeted advertising (Adware)

Scareware: masquerades as useful software but when executed will infect or completely destroy system

Rootkits: designed to gain root access (admin privileges) and then can control system

DDoS: (distributed denial-of-service) seeks to overload the bandwidth and resources of a system by bombarding it with connections all at once (from multiple distributed locations) to disable or crash the system

DATA PRIVACY



Unlike much of the data we have been discussing to date which is “open” and free to use by anyone who can access it, the data and information we wish to keep secure is **private**

Protecting private information is a **business requirement**, and in many cases also an **ethical** and **legal** requirement

Private data must be kept secure:

- at rest (stored) and when being transmitted
- when being processed (by programs or people)
- when being disposed (deleted) or devices it is stored on are being disposed

PERSONAL DATA



When you give your personal data to a person or organisation that organisation has a legal duty to keep that data private and safe

Data protection ensures that legally your personal data should be:

- factually correct.
- only available to those who should have it.
- only used for stated purposes.

(www.dataprotection.ie)

OTHER RELATED ISSUES ...

- Is all data collected necessary?
- How long is data preserved?
- Is it shared/given to others?
- When should it be disposed?
- How should it be disposed?

GENERAL DATA PROTECTION REGULATION (GDPR)

“The General Data Protection Regulation is a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA). It also addresses the export of personal data outside the EU and EEA areas. The GDPR aims primarily to give control to individuals over their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU.”

DATA PROTECTION

by design and by default (GDPR)

- highest-possible privacy settings should be used by default
- data should **not be** available publicly without explicit, informed consent
- data cannot be used to identify a subject without additional information stored separately
- no personal data may be processed unless it is done under a lawful basis specified by the regulation or unless the data controller or processor has received an unambiguous and individualized affirmation of consent from the data subject
- the data subject has the right to revoke this consent at any time

IN THE NEWS



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute

Help
Learn to edit
Community portal
Recent changes
Upload file

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export



THE IRISH TIMES

NEWS

EDUCATION

STUDENT HUB

2ND LEVEL HUB

PARENTS



Attempted cyberattack causes disruption at NUI Galway

Online lectures impacted as university disconnects network from wider internet

Thu, Sep 30, 2021, 13:50

Not logged in Talk Contributions Create account Log in

Article Talk

Read Edit View history

Search Wikipedia

Health Service Executive ransomware attack

From Wikipedia, the free encyclopedia

On 14 May 2021, the **Health Service Executive (HSE)** of Ireland suffered a major ransomware cyberattack which caused all of its **IT systems** nationwide to be shut down.^{[1][2][3][4]}

It was the most significant **cybercrime attack** on an **Irish state agency** and the largest known attack against a health service computer system.^{[5][6]} Bloomberg News reported that the attackers used the **Conti ransomware**.^[7] The group responsible was identified as a criminal gang known as **Wizard Spider**, believed to be operating from Russia.^{[8][9][10]} The same group is believed to have attacked the **Department of Health** with a similar cyberattack.

On 19 May, the **Financial Times** reviewed private data for twelve individuals which had appeared online as a result of the breach.^[11] On 28 May, the HSE confirmed confidential medical information for 520 patients, as well as corporate documents were published online.^[12]

On 23 June, it was confirmed that at least three quarters of the HSE's IT systems had been restored, with 70% of computer devices back in use.^[13] By September, over 95% of the data had been restored.^[14]

Health Service Executive ransomware attack

Date 14 May 2021

Location Ireland

Type Ransomware

Target Health Service Executive

Outcome

Suspects



Alerts were raised within the health service over eight weeks that the IT system might be compromised, but the significance of the alerts was not identified at the time. Photograph: iStock



The opening of a malicious Microsoft Excel file attached to a phishing email led to the cyber attack that crippled the national health service earlier this year, according to a report on the incident published on Friday.



THREE ATTRIBUTES OF SECURE DATA:

CIA TRIAD:

- Confidentiality
 - Integrity
 - Availability
-
- The focus of Information security approaches is the balanced protection of the CIA triad.
 - Information security requires a culture of “**continual improvement**”.

INFORMATION SECURITY DEFINITION *revisited*

"The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability".

1. CONFIDENTIALITY: Information that is secret should stay secret

- Only people authorised to access the information may do so
- **Encryption** and **Access Controls** used to protect data and information confidentiality
- Threats to confidentiality include malware, intruders, device theft, password theft, insecure networks, systems and apps, poorly administered systems, emails sent to wrong people, wrong attachments in emails, lack of laws and regulations, whistle-blowers.

2. INTEGRITY

Integrity relates to the trustworthiness, origin, completeness, and correctness of information

This means that data should not be modified in an unauthorized or undetected manner.^[1]

To protect integrity:

- Prevent unauthorized modification of information (via Access Controls)
- Detect unauthorised modifications.

3. AVAILABILITY

Availability relates to ensuring that authorised users of information and systems can access and use them when they want to

Threats to availability:

- Natural and human disasters can affect availability
- Malicious attacks against availability are known as denial of service (DoS or DDoS) attacks and prevent availability of systems and access to data
- Other security breaches

High Availability Systems aim to remain available at all times – even through power outages, hardware failures, and regular system upgrades (e.g., Google email)

IN THE NEWS

RTE

NEWS SPORT ENTERTAINMENT BUSINESS LIFESTYLE CULTURE PLAYER TV RADIO

NEWS > BUSINESS > Business of Climate Future of Work Brexit Watch and Listen Your Money Technology

Bank of Ireland fined €24.5m by Central Bank for regulatory breaches

Updated / Thursday, 2 Dec 2021 20:11



This is the largest ever fine in this area of enforcement in the Central Bank's history



By **Will Goodbody**
Business Editor

Bank of Ireland has been fined €24.5m by the Central Bank for regulatory breaches related to its IT systems and related internal controls.

The regulator found that Bank of Ireland failed to have a robust framework in place to ensure continuity of service for it and its customers in the event of a significant IT disruption.

The lender also didn't have effective internal controls in place to identify such issues and ensure they were brought to the attention

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ON THEIR OWN)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

NOTE: “THE WEAKEST LINK”

The strength of any system is no greater than its weakest link.

If different security mechanisms are enforced for each component of an information system then there is “**defence in depth**”

- i.e., the build up and layering of security mechanisms, so if one fails, there are still other mechanisms in place

Research, and experience, has shown that the most vulnerable point in most information systems is the **human user, operator, designer, or other human**

APPROACH 1: IDENTIFICATION, AUTHENTICATE, AUTHORIZE SEQUENCE

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

Identification

- Identifying the who (person) or what (entity)
- Usually, using some unique *username* or code
- Must be locally unique and possibly globally unique so that access control may be enforced and accountability established
 - e.g., email addresses, twitter handler, alias.

APPROACH 1: IDENTIFICATION, AUTHENTICATE, AUTHORIZE SEQUENCE

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

Authentication

Verifies the authenticity of the identity declared at the identification stage:

Three methods of authentication:

- What you know
- What you have
- What you are

... WHAT YOU KNOW



- Passwords, passphrases, secret codes, and personal identification numbers (PINs).
- Most popular
- Low cost and easy to implement
- Often stored insecurely
- Often easy to guess ...

PROBLEMS WITH PASSWORDS

Human-generated passwords are often very easy to guess by a machine because:

- often short and use real words found in dictionaries
- use “obvious” passwords (name, “password”, “abc123”)
- same password used across multiple systems
- a written (unsecure) version of the password might exist
- prone to phishing – spoof emails or websites that trick people in to entering their valid password (and other private data)

PASSWORD MANAGERS

- Password managers are applications that **generate** strong passwords as well as securely **store** passwords to systems you use
- Only need to remember a single (hopefully strong!) password to log in to the password manager
- Data may be stored locally or on the cloud
- Very important for managing passwords
- Examples:



1Password



SplashID

LastPass...|

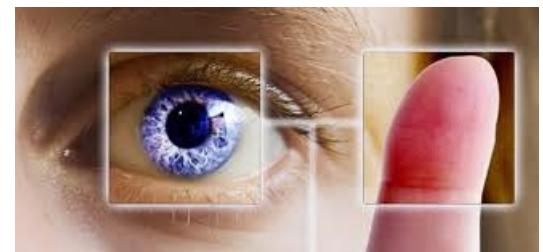
AUTHENTICATION

What you have ...

- Keys, swipe cards, smart cards, bank cards, phones, token generator, etc.
- Higher costs and problems when lost, stolen, damaged etc.

What you are ...

- Biometric authentication methods
- Examples:
 - Face, fingerprint, iris, and retina recognition
 - voice and signature recognition
- Included in IDs, Passports, smart phones and many computing devices



MULTI-FACTOR (STRONG) AUTHENTICATION

The trend in most apps and secure systems is to provide more than one type of authentication to overcome the limitations with password-only based authentication.

Multi-factor authentication requires the user to present two or more codes (evidence/factors) to prove who they are.

Many sites, including Google or Facebook, only ask for the second factor when you sign in from a new device (or using a different browser).

2FA: TWO FACTOR AUTHENTICATION

- Two-factor authentication (2FA) is most common nowadays
- 2FA is generally supported with “what a user has”, e.g., a smartphone, personal security key, 3rd party application
- Often involves the generation of a Time-based One-Time Password (TOTP) which is sent via SMS or generated on the 3rd party app or device.
- “Push” notifications, in conjunction with an app, are also used (e.g. banking apps in particular)

Sample 3rd part Authentication Apps: Authy; Microsoft Authenticator; CISCO Duo; Google Authenticator; LastPass Authenticator

Time-based One-Time Password (TOTP)

Codes are generated using an algorithm and each code lasts a very short period of time.

For TOTP on 3rd part applications or for physical devices, codes are generated based on a number assigned to the device or app and the current time. In this approach, only the local physical device or app has the code, which makes them more secure than text-message or email codes.

PUSH NOTIFICATIONS

A notification is sent to phone and user taps/swipes to approve the login or payment

Sometimes push notifications could ask user to match a code

Push notifications are easier to use and more secure than TOTP, but are currently not available for many sites.

USAGE

Two-factor authentication is recommended by the National Institute of Standards and Technology (NIST) to secure online accounts that deal with personal information, the collection of personal information, or the maintenance of personal information.

Ideally, Two-factor authentication should be enabled on password managers, email, cloud backup services, bank accounts, social media profiles, chat apps, and any app with your health and fitness data.

Note: Issue with **digital exclusion** for those without smartphones, internet access, and/or for those with disabilities (e.g., sight) which may prevent them from using these approaches.

AUTHORIZATION

Once identification and authentication have been complete then access is authorized

Also referred to as rights, privileges, or permissions

- Ensures that person has the required rights to perform the tasks they need to while being prevented from performing other tasks
- Enforces access control and accountability

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

ACCOUNTABILITY

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

Systems must have accountability

- i.e., being able to trace actions and events back in time to the users, systems, or processes that performed them, to establish responsibility for security breaches

Mainly provided by logs and an audit trail

ACCESS CONTROL: POLICIES AND CODE OF CONDUCT:

The foundation of access control is the identification and authentication sequence - This ensures that access to protected information is restricted to people who are authorised to access it.

Computer programs, and computers, that process the information, must also be authorised - this requires that mechanisms be in place to control the access to protected information.

The level of access control required should relate to the value of the information being protected, i.e., the more sensitive or valuable the information the stronger the control mechanisms need to be.

POLICIES/CODE OF CONDUCT:

The level of access control required is specified as Policies and Code of Conduct.

These involve:

- assigning a security classification to data
- specifying an access control policy
- specifying a “handling” rule

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

Assigning Security Classification Examples:

In Business sector:

- Public
- Sensitive
- Private
- Confidential

In the government sector:

- Unclassified
- Sensitive But Unclassified
- Restricted
- Confidential
- Secret
- Top Secret

MORE GENERALLY ...

Traffic Light Protocol (TLP) classification and handling can be defined as follows:

- White: unlimited – distribute freely
- Green: community wide distribution
- Amber: limited “need-to-know” distribution
- Red: Very restricted - for named people only



ACCESS CONTROL MODELS

The *principle of least privilege* stipulates:

“Do not give any more privileges than absolutely necessary to do the required job.”

Three main access control models exist:

- *discretionary* access control model (DAC)
- *mandatory* access control model (MAC)
- *role-based* access control model (RBAC)

ACCESS CONTROL MODELS

- **DAC:** individual users can specify the security aspects of an object
- **MAC:** opposite of DAC: security policy is centrally controlled and specified on objects by a security policy administrator and users do not have the ability to override the policy
- **RBAC:** defined around roles and privileges where there exist many people in an organisation; users are not assigned permissions directly but acquire them through their roles

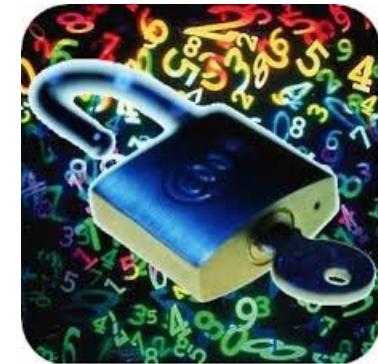
ENCRYPTION

SECURITY APPROACHES (NOT NECESSARILY SOLUTIONS ...)

1. Identification, Authenticate, Authorize Sequence
2. Accountability
3. Policies/Code of Conduct: specification of the security properties a system should have and how people should treat data
4. Encryption: transforming data into something which cannot be understood, e.g., cryptography

Cryptography:

The art and science of keeping messages secret



[home](#) > [UK](#)

the guardian
Winner of the Pulitzer prize

[UK](#) [world](#) [sport](#) [football](#) [opinion](#) [culture](#) [economy](#) [lifestyle](#) [fashion](#) [environment](#) [tech](#) [money](#) [travel](#) [≡ browse all sections](#)

UK news

Security services capable of bypassing encryption, draft code reveals

Home Office code of practice spells out rules and safeguards surrounding use of computer hacking outside UK

Alan Travis, home affairs editor

Friday 6 February 2015 18.42 GMT

A dark, abstract background featuring a grid of binary digits (0s and 1s) arranged in a hexagonal pattern, creating a sense of digital complexity and security.

Most popular

WHY REQUIRED?

- The digital equivalent of envelopes and locked filing cabinets
- Particularly important for:
 - Internet ... medical, law and financial data, banking, passwords
 - Online shopping and transactions
 - Military and Diplomatic communications
 - Human Rights Organisations
 - Journalists

TERMINOLOGY

Plaintext = original message

Ciphertext = encrypted message

Encryption:

- transform plaintext to ciphertext usually using a key

Decryption:

- transform ciphertext to plaintext (with key)



CRYPTOGRAPHY WORKS ON MANY LEVELS:

1. Algorithms
2. Protocols (built on the algorithms)
3. Applications (built on the protocols)

Some examples in everyday use:

- SSL .. secure socket layer
- HTTPS ... secure HTTP
- OpenPGP ... Pretty Good Privacy
- SFTP

TWO CLASSES OF CRYPTO SYSTEMS:

1. Symmetric

One key: secret/private key

Use same key for encryption and decryption

Can be divided into:

- *stream ciphers* - encrypt a single bit at a time
- *block ciphers* - take a number of bits and encrypt them as a unit

2. Asymmetric: Public Key

Two keys: public key and private key

Uses a different key for encryption and decryption

Decryption key “cannot” be derived from the encryption key

SYMMETRIC BLOCK CIPHER CRYPTOSYSTEMS

DES (data encryption standard)

3DES

Blowfish, Twofish, Threefish

AES (advanced encryption standard)

DATA ENCRYPTION STANDARD (DES)

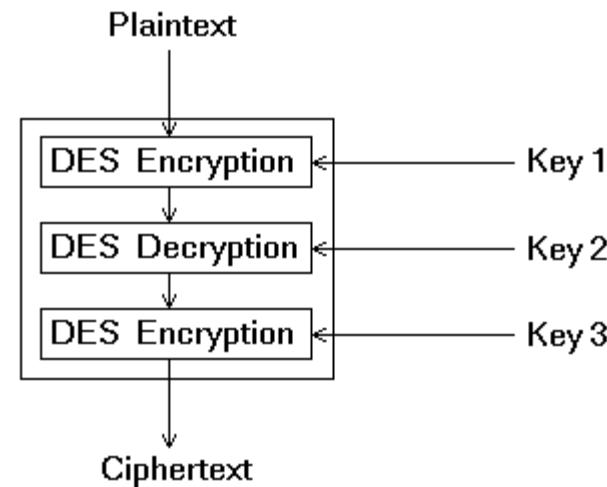
- Developed in IBM mid 1970s and later used by US government
- Same algorithm and key used for encryption and decryption
- Consists of 16 “rounds” (substitutions and transpositions) of operations that mix the data and key together
- Data is encrypted and decrypted in 64-bit chunks
- Goal is to completely scramble data
- However in late 1990s, code was cracked in a few hours

TRIPLE DES (TDES)

Uses 3 keys

3 times slower than regular DES

Billions of times more secure if used properly



AES

- Established as a standard by National Institute of Standards and Technology in 2001
- Used worldwide today by government and private sector
- Standard encryption approach for “data at rest”
- Fast encryption and decryption

HOW TO TRANSMIT KEY FOR SYMMETRIC CRYPTO SYSTEMS?

- Meet personally
- By phone
- Electronically
- Using asymmetric cryptography (PGP)

One of the weak links in symmetric cryptography is the transmission of keys between the sender and the receiver

Public Key cryptography avoids having to send (private) keys

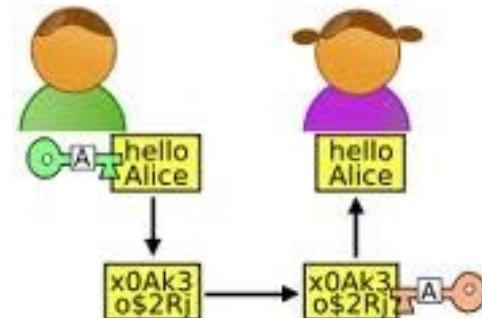
PUBLIC KEY CRYPTOGRAPHY

"The face of cryptography was radically altered when Diffie and Hellman invented an entirely new type of cryptography, called public key. At the heart of this concept is the idea of using a one-way function for encryption."

from "Algebraic Aspects of Cryptography" by Neal Koblitz

Public key cryptography is an encryption and decryption technique that enables entities to securely communicate on an insecure public network, and reliably verify the identity of an entity via digital signatures.

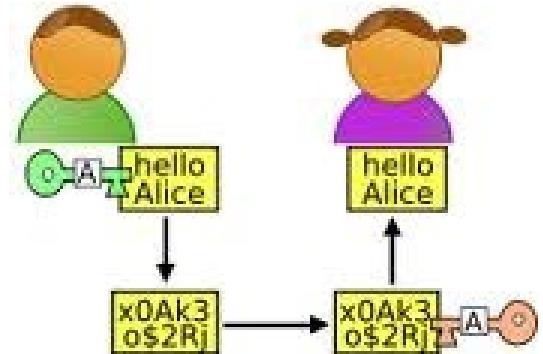
ALICE AND BOB ...



- One box, Alice makes freely available an open lock (or many locks) but not the key
- Bob puts something confidential in to a box to send to Alice and uses Alice's lock to lock it
- Then box is sent to Alice
- Only Alice has key and only Alice can open it



ASYMMETRIC CRYPTOSYSTEMS



- Invented in 1976
- Each person gets a pair of keys ... public and private
- The plaintext message is encrypted using the public key
- The private key is used to decrypt ciphertext message
- Example: RSA
- Security arises from mathematical difficulty in obtaining private key from public key (feature of one-way functions)
- Can be used for authentication as well as privacy

ASIDE: ONE-WAY (“Trap door”) FUNCTIONS

A function f which is easy to apply to an input number x to give $f(x)$ but difficult/impossible to recover the original number x , knowing only $f(x)$.

Other Maths used:

- Euclidean Algorithm
- Fermat’s Theorem
- Elliptical Curves

Will cover this in more detail in MA160/MA190

PKI: Public Key Infrastructure *for key management*

Public Key infrastructure (PKI) is a system for the creation, storage, and distribution of digital certificates for public-key encryption to verify that a certain public key belongs to a certain entity.

PKI creates, manages, distributes, stores and, if necessary, revokes digital certificates.

PRETTY GOOD PRIVACY (PGP)

- Processing of RSA requires large amounts of computing power.
- With the original systems like DES this power was not needed.
- Pretty Good Privacy combines some of the best features of both the RSA public key cryptosystem and a standard symmetric cryptosystem

STEPS FOR ENCRYPTION:

1. Message m is first compressed thus reducing the patterns found in natural languages
2. A one-time-only secret key is created randomly
3. This key is used to encrypt message (using 3DES or AES for example)
4. Symmetric key is then encrypted using public key
5. This public key-encrypted symmetric key is transmitted along with the encrypted message

DIGITAL SIGNATURES

Signatures have been used for centuries to authenticate messages

RSA can be used for digital signatures

Interchanges the roles of public and private keys, such that a message can be encrypted with the private key and decrypted with the public key

“Digital signatures are about delivering a service aimed at data integrity and are not about encryption”

DIGITAL SIGNATURE APPROACH

Approach:

- 1. Publish RSA public key as usual
- 2. Encrypt message m with private key to give y , and present y as signed version of m

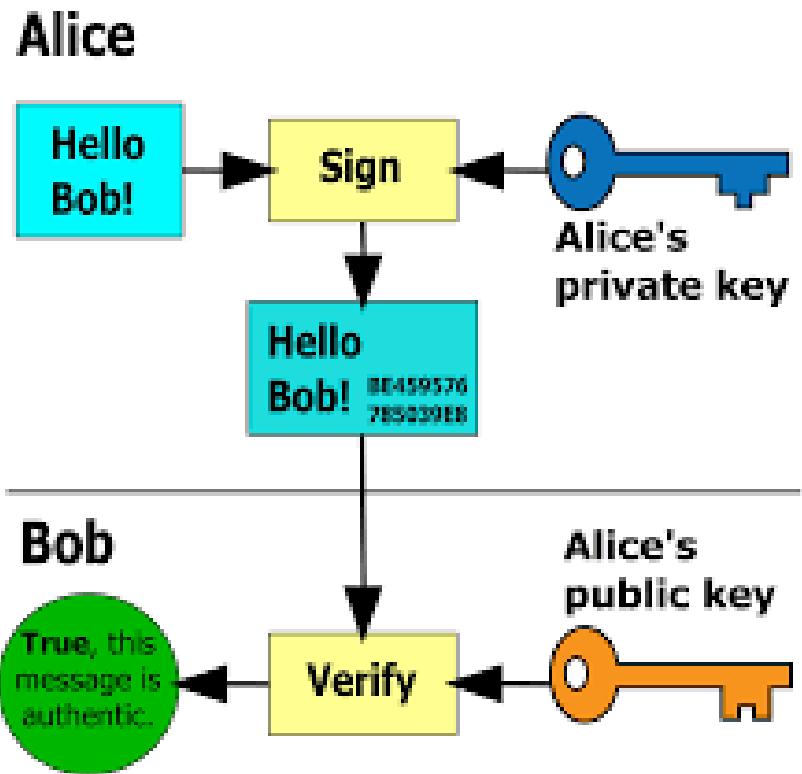
Idea:

Anyone can decrypt message y using the public key

If the decrypted signature makes sense, then the signature is authenticated.

Essentially, a person uses their own private key to encrypt their signature and sends this as part of message

The person's public key is used to decrypt the signature part – if the resulting message is the sender's signature then it authenticates the message as only the sender had access to the private key to encrypt the signature.



from Wikipedia

HTTPS

A secure version of HTTP ([HyperText Transfer Protocol](#))

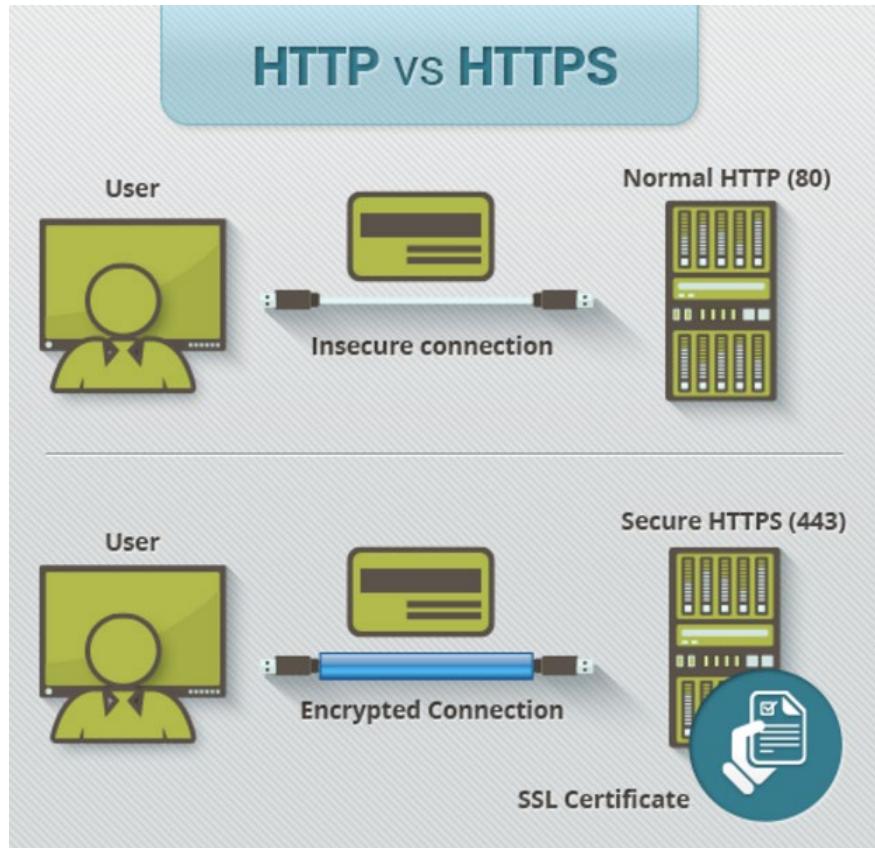
Used to [protect](#) highly confidential online transactions like online banking and online shopping data

HTTPS pages typically use one of two secure protocols to encrypt communications:

- SSL (Secure Sockets Layer) or
- TLS (Transport Layer Security)

Both the TLS and SSL protocols use '[asymmetric](#)' Public Key Infrastructure (PKI) system

HTTP VS HTTPS



From: <https://www.instantssl.com/ssl-certificate-products/https.html>

Sample HTTPS “handshake” approach: STEPS:

1. Client → Server

“hello” and sends details of encryption and SSL it uses

2. Server → Client

“hello” and sends SSL certificate containing Public Key and Digital Signature

3. Client → Server:

message M encrypted with randomly created private key, P and send

encrypt private key P with server’s public key and send

ATTACKS ON ASYMMETRIC CRYPTOSYSTEMS

Knowing Private Key

Advances in factoring ... quantum computing etc.

Weakness in Implementation (**side-attacks/sabotaging**: “back doors” in design or implementation or in operating system)

SUMMARY

- Data and information system security is one of the most reported aspects of Computing in the media
- A “story” of “villains” and “spies” - with whistle blowers, espionage, government secrets, government influence, etc.
- In the Internet age, the age of Cloud computing, the age of personal devices, and advances in quantum computing, challenges and controversies only set to continue ...